

## ABSTRACT

Title of Dissertation: SYNTHESIS AND CHARACTERIZATION  
OF ENERGETIC NANOMATERIALS WITH  
TUNABLE REACTIVITY FOR  
PROPULSION APPLICATIONS

Dylan Jacob Kline, Doctor of Philosophy in  
Chemical Engineering, 2020

Dissertation directed by: Professor Michael R. Zachariah, Department of  
Chemistry & Biochemistry and Department of  
Chemical & Biomolecular Engineering

Combustion is the world's leading energy conversion method in which a fuel and oxidizer react and release energy, typically in the form of heat. Energetic materials (propellants, pyrotechnics, and explosives) have combustion reactions that are so fast that they are generally limited by how quickly the fuel and oxidizer can reach each other. Recent research has employed nanomaterials to reduce the distance between reactants to increase energy release rates. This dissertation attempts to uncover and quantify structure-function relationships in energetic nanomaterials by modifying chemical and physical properties of the materials and characterizing the observed changes using new diagnostic tools.

This dissertation begins with the development of diagnostic tools that can capture the dynamics of energetic material combustion using a high-speed color camera to measure temperature. This tool has also been modified into a high-speed microscope

that allows for spatial and temperature measurements at microscale length ( $\mu\text{m}$ ) and time ( $\mu\text{s}$ ) scales.

Changes to chemical formula have been explored for energetic nanomaterial systems, though visualization of the reaction dynamics limited detailed results on reaction mechanisms. The first study performed here probed the role of gas generation vs. thermal effects in energy release rate where it was found that combustion inefficiencies from reactive sintering could be mitigated by introducing a gas-generating oxidizer. To explore combustion improvements in the fuel, a metal fuel nanoparticle manufacturing method was explored, though the combustion performance was again limited by reactive sintering. Another effort to reduce reactive sintering with a gas generator proved successful, but also unveiled the importance of different heat transfer mechanisms for propagation.

The role of physical architecture on propellant combustion was also investigated to improve efficiency and versatility in solid propellants. It was found that addition of a poor thermal conductor to a propellant mixture increased the propagation rate of the material and this was attributed to the result increase in burning surface area resulting from inhomogeneous heat transfer. Lastly, this dissertation explores a method to remotely ignite materials using microwaves and titanium nanoparticles. This work sets the stage for a remotely staged solid propellant architecture that would provide control over solid propellant combustion in-operando.

SYNTHESIS AND CHARACTERIZATION OF ENERGETIC  
NANOMATERIALS WITH TUNABLE REACTIVITY FOR PROPULSION  
APPLICATIONS

by

Dylan Jacob Kline

Dissertation submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park, in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
2020

Advisory Committee:

Professor Michael R. Zachariah, Advisor & Co-Chair  
Professor Dongxia Liu, Co-Chair  
Professor Akua Asa-Awuku  
Professor Kenneth T. Kiger  
Professor Taylor Woehl  
Professor Peter B. Sunderland, Dean's Representative

© Copyright by  
Dylan Jacob Kline  
2020



## Dedication

To my family, Cristine, Lawrence, and Elayna, for encouraging me to push myself out of my comfort zone, for bringing home to me wherever I was in this country, for fielding a never-ending list of random questions on random topics at random times, and for inspiring me to be the best version of myself. To Josh, Jordan, and Thea, for being my rocks in times of need, for providing distractions even when I didn't need them, and for being my family through everything. To my friends from UMD and NASA, Emily, David, Kyle, Lauren, Kim, Kätchen, Ricky, Miles, Shannon, Evelyn, Matt, Jillian, Elaine, Nithin, and Nicole, for the countless memories, for going on adventures in new places with me, and for embracing my quirks with open arms. And to Gabi, Brenna, Adam, Katie, Stephen, and Stephen for giving me life outside of the lab, for making time for visits and phone calls, and for being there irrespective of the distance between us. There are too many people to thank for making this experience special and for the unwavering support – I am only here because of you all. As they say, “it takes a village,” and thank you for being my village.

“Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning.” – Winston Churchill.

## Acknowledgements

This work has only been possible with the incredible amount of support I have received by my advisors, labmates, collaborators, and sponsors.

First off, I would like to thank my advisor, Dr. Michael Zachariah, for incredible mentorship at every step in the process over the past four years as part of the Zachariah Group. Dr. Zachariah has had every role imaginable as a mentor ranging from research advisor to friend since I started my degree and I cannot thank him enough for giving me the opportunity to work in such an exciting, collaborative workspace. Surely, there have been some trials and tribulations (like a cross-country move), but this has been such a rewarding and enriching experience. I couldn't imagine not being here. Thank you for everything, truly.

Additional thanks are also in order for all of my collaborators that have been part of the Zachariah Group over the past four years. In particular, I'd like to thank my mentors at the start of this process, Dr. Rohit Jacob and Dr. Phil Guerieri, and my labmates, Dr. Haiyang Wang, Dr. Jeffery DeLisio, Dr. Xizheng Wang, Dr. Tao Wu, Dr. Scott Holdren, Dr. Kevin Yang, Dr. Will Gibbons, Dr. George Mulholland, Miles Rehwoldt, Zaira Alibay, Pankaj Ghildiyal, Feiyu Xu, Prithwish Biswas, Yujie Wang, Erik Hagen, and Matthew Krock. You all have provided invaluable scientific discussion, embraced all of the collaborative efforts with open arms, and have been part of some of the greatest memories I could ever ask for. This group has always felt like a family to me and has made me excited to come to school every day because you just don't know what is going to happen. I am so lucky to have something that makes saying goodbye so hard.

There have been other research collaborators that have provided invaluable insight into experiments and work in my dissertation – notably Charles Turner, Shannon McDonnell, Yonggang Yao, Professor Liangbing Hu, Spencer Hamilton, Alexander Idrogo-Lam, Carlos Rodriguez, Sara Barron, Greg Fritz, and Aaron Mason. I appreciate all of the conversations we have had and cannot express how thankful I am for taking the time to work with me.

Thank you to my advisory committee – Professor Akua Asa-Awuku, Professor Kenneth Kiger, Professor Dongxia Liu, Professor Peter Sunderland, and Professor Taylor Woehl – for sharing their knowledge with me and giving me resources to become a better researcher.

This work was made possible with support from the National Science Foundation, Army Research Office, Air Force Office of Scientific Research, Office of Naval Research, and Defense Threat Reduction Agency. The University of Maryland has also provided financial support as part of the Jan & Anneke Sengers Fellowship and Ann G. Wiley Dissertation Fellowship. Thank you to all of these sponsors for continuing investment in education and academic research.

# Table of Contents

Dedication .....	ii
Acknowledgements .....	iii
Table of Contents .....	v
List of Tables .....	xi
List of Figures .....	xiii
List of Abbreviations .....	xxii
Chapter 1 : Introduction .....	1
1.1 Combustion and energetic Materials .....	1
1.1.1 History of energetic materials .....	2
1.1.2 Chemical reaction kinetics and transport phenomena basics .....	5
1.1.3 Combustion performance characterization techniques .....	11
1.1.4 Homogeneous vs. heterogeneous energetic materials .....	14
1.2 Energetic nanomaterials .....	17
1.2.1 Mechanisms of energetic nanomaterial combustion .....	18
1.2.2 New methods to tailor energetic nanomaterial combustion .....	28
1.3 Motivation and research outline .....	31
1.3.1 Developing diagnostic tools for characterizing energetic material combustion .....	32
1.3.2 Probing the role of chemical formulation energetic material combustion .....	32
1.3.3 Probing the role of physical architecture on energetic material combustion .....	33
Chapter 2 : High-speed 2-dimensional temperature measurements of nanothermite composites: Probing thermal vs. gas generation effects .....	35
2.1 Summary .....	35
2.2 Introduction .....	36
2.3 Experimental .....	39
2.3.1 Materials and sample preparation .....	39
2.3.2 Constant volume combustion cell .....	39
2.3.3 High speed 32-channel spectrometer .....	41
2.3.4 Hot-wire ignition tests for spatiotemporal temperature maps .....	44
2.4 Results .....	47
2.4.1 Al/CuO nanothermite tests in pressure cell .....	47
2.4.2 Al/Fe <sub>2</sub> O <sub>3</sub> and Al/WO <sub>3</sub> nanothermite tests in pressure cell .....	50
2.4.3 Qualitative observation of reaction dynamics with camera .....	53
2.5 Mechanism .....	56
2.6 Conclusions .....	60
2.7 Acknowledgements .....	61
2.8 Supplemental Information .....	61
2.8.1 Calibration of a high-speed spetcrometer .....	61
2.8.2 Calibration of a high-speed color camera pyrometer .....	62
Chapter 3 : Combustion dynamics of nanoaluminum-laden reduced graphene oxide fabricated by laser pulse heating .....	66
3.1 Summary .....	66

3.2 Introduction.....	66
3.3 Methods.....	69
3.3.1 Preparation of metallized reduced graphene oxide sheets .....	69
3.3.2 Laser-induced thermal shock of metal/rGO sheets .....	70
3.3.3 Temperature measurements .....	71
3.3.4 Particle imaging .....	72
3.3.5 Thermogravimetric analysis/differential scanning calorimetry .....	72
3.4 Results and discussion .....	72
3.4.1 Characterization of original rGO/Al films .....	72
3.4.2 In-situ nanoparticle generation .....	74
3.4.3 Combustion characteristics of rGO/Al films .....	78
3.5 Conclusions.....	82
3.6 Acknowledgements.....	83
3.7 Supplemental Information .....	83
3.7.1 Characterization of original rGO/Al films.....	83
3.7.2 In-situ nanoparticle generation for different loadings .....	84
3.7.3 Combustion characteristics of rGO/Al films .....	85
Chapter 4 : <i>In-operando</i> high-speed microscopy and thermometry of reaction propagation and sintering in a nanocomposite.....	86
4.1 Summary .....	86
4.2 Introduction.....	86
4.3 Results.....	90
4.3.1 Evaluating the reaction zone.....	90
4.3.2 Mapping optical to electron microscopy.....	93
4.4 Discussion .....	98
4.5 Methods.....	101
4.5.1 Chemicals and precursors .....	101
4.5.2 3D-printing on glass slides.....	102
4.5.3 Additional information about high-speed microscope imaging system ..	102
4.5.4 Color video pyrometry .....	102
4.5.5 Morphology characterization .....	103
4.6 Acknowledgements.....	103
4.7 Supplementary Information .....	104
Chapter 5 : Experimental observation of the heat transfer mechanisms that drive propagation in 3D printed reactive materials.....	108
5.1 Summary .....	108
5.2 Introduction.....	109
5.3 Methods.....	111
5.3.1 Chemicals.....	111
5.3.2 Precursor preparation .....	112
5.3.3 Direct-writing of burn sticks .....	113
5.3.4 Macro-scale burn tests .....	113
5.3.5 High-speed in-operando microscopy/thermometry and particle tracking	114
5.3.6 Morphology characterization .....	115
5.4 Results.....	115
5.4.1 Materials .....	115

5.4.2 Macro-scale burn tests .....	116
5.4.3 High-speed in-operando microscopy .....	118
5.4.4 Post-reaction forensics .....	121
5.5 Discussion .....	123
5.5.1 Conductive heat transfer estimates .....	124
5.5.2 Radiative heat transfer estimates .....	125
5.5.3 Convective heat transfer estimates .....	126
5.5.4 Metal vapor condensation estimates .....	127
5.5.5 Advective heat transfer .....	129
5.6 Conclusions .....	130
5.7 Acknowledgements .....	131
5.8 Supplemental Information .....	132
Chapter 6 : <i>In-operando</i> thermophysical properties and kinetics measurements of Al-Zr-C composites .....	137
6.1 Summary .....	137
6.2 Introduction .....	137
6.3 Methods .....	141
6.3.1 Sample preparation .....	141
6.3.2 Morphology characterization .....	142
6.3.3 High-speed videography/pyrometry .....	143
6.3.4 Determination of transport and kinetic properties .....	144
6.3.5 Thermophysical property estimations .....	146
6.3.6 Kinetic parameter estimation .....	149
6.4 Results .....	149
6.4.1 Propagation characterization .....	149
6.4.2 Temperature profiles .....	151
6.4.3 Extraction of thermophysical properties .....	154
6.4.4 Kinetic parameters .....	156
6.5 Conclusions .....	158
6.6 Acknowledgements .....	159
Supplemental Information .....	160
Chapter 7 : Why does adding a poor thermal conductor increase propagation rate in solid propellants? .....	162
7.1 Summary .....	162
7.2 Introduction .....	162
7.3 Results .....	165
7.3.1 Macroscale combustion tests .....	165
7.3.2 Microscale combustion tests .....	166
7.4 Discussion .....	170
7.5 Conclusions .....	171
7.6 Acknowledgements .....	171
7.7 Supplemental Information .....	172
Chapter 8 : Numerically evaluating energetic composite flame propagation with thermally conductive, high aspect ratio fillers .....	174
8.1 Summary .....	174
8.2 Introduction .....	174

8.2.1 Thermal transport and chemistry .....	177
8.2.2 Physical layout .....	180
8.2.3 Model analysis and parameters .....	182
8.3 Results .....	183
8.3.1 Role of rod aspect ratio in combustion performance .....	183
8.3.2 Role of rod thermal conductivity in flame propagation .....	189
8.4 Conclusions .....	192
8.5 Acknowledgements .....	193
8.6 Supplemental Information .....	193
8.6.1 Numerical methods .....	193
8.6.2 Additional figures .....	198
Chapter 9 : Spatially focused microwave ignition of metallized energetic materials .....	200
9.1 Summary .....	200
9.2 Introduction .....	200
9.3 Methods .....	204
9.3.1 Mie theory calculations .....	204
9.3.2 Materials .....	204
9.3.3 Film characterization .....	205
9.3.4 Burn rate characterization .....	207
9.3.5 Focused microwave ignition .....	207
9.4 Results and Discussion .....	209
9.4.1 Microwave absorption of metals and materials selection .....	209
9.4.2 Microwave sensitivity of metal powders .....	215
9.4.3 Microwave sensitivity of metallized propellants .....	217
9.4.4 Controlled-environment burn characteristics of metallized propellants ..	219
9.4.5 Architecturing of a microwave-initiated solid propellant .....	221
9.5 Conclusions .....	224
9.6 Acknowledgements .....	225
9.7 Supplemental Information .....	226
Chapter 10 : Summary .....	231
10.1 Conclusions .....	231
10.2 Recommendations for future work .....	236
10.2.1 Exploiting additive geometry for enhanced heat transfer in energetic materials .....	236
10.2.2 Microwave preheating as a method to throttle a solid propellant in-operando .....	238
Appendices .....	240
Appendix A: Experimental description and useful information for color camera pyrometry .....	240
A.1. Introduction .....	240
A.2. Descriptions of available cameras .....	240
A.3. Procedure for performing color camera pyrometry experiments .....	241
A.4. Recommendations for performing pyrometry experiments .....	242
Appendix B: Calibrating a color camera pyrometer .....	245
B.1. Procedure for calibrating camera pyrometer .....	245

B.2. Program for extracting single channel values for calibration ratio calculations .....	248
B.3. Program for calculating predicted ratios for color camera pyrometer .....	249
Appendix C: MatCine – a multiplatform MATLAB program to extract raw data from Phantom Camera “.cine” video formats.....	254
C.1. Introduction .....	254
C.2. Running MatCine .....	255
C.3. MATLAB code .....	255
Appendix D: pyroCam – A MATLAB script for color camera pyrometry .....	262
D.1. Running pyroCam .....	262
D.2. Description of input and output .....	263
Input .....	263
Output .....	272
D.3. Variable description .....	273
D.4. Code structure .....	278
D.5. Summary .....	279
D.6. Version history .....	280
D.7. MATLAB code .....	281
Appendix E: Experimental description for high-speed microscopy .....	297
E.1. Introduction .....	297
E.2. Descriptions of available cameras/lenses .....	298
E.3. Procedures for performing high-speed microscopy .....	298
Appendix F: MATLAB script for extracting flame front surface area using high-speed microscopy.....	303
F.1. Running flame front extraction code .....	303
F.2. Description of input and output .....	304
Input .....	304
Output .....	308
F.3. Variable description.....	309
F.4. Code structure.....	312
F.5. Summary.....	314
F.6. Version history.....	314
F.7. MATLAB code.....	315
Appendix G: T-Jump Nano – A multi-signal pulse generator for precision triggering and heating.....	328
G.1. Introduction.....	328
G.2. Project purpose.....	328
G.3. T-Jump Nano.....	329
Initial design and breadboard prototype .....	329
Printed circuit board design .....	336
G.4. Other uses.....	339
G.5. Acknowledgements.....	339
Appendix H: Arduino scripts for the T-Jump Nano in different configurations .....	340
H.1. T-Jump Nano Time of Flight Mass Spectrometry (TOFMS) .....	340
H.2. T-Jump Nano Microscope.....	343
H.3. T-Jump Nano for Pressure Cell and Spectrometry Measurements .....	345



H.4. T-Jump Nano TTL .....	347
Appendix I: T-Jump Nano Ultimate – A better multi-signal pulse generator for precision triggering and heating.....	349
I.1. Introduction .....	349
I.2. Project purpose .....	349
I.3. T-Jump Nano Ultimate .....	350
Printed circuit board design .....	350
G.5. Acknowledgements .....	353
Appendix J: Arduino scripts for the T-Jump Nano Ultimate.....	354
Appendix K: Spatially focused microwave needle apparatus operation .....	359
K.1. Introduction, project purpose, and background .....	359
K.2. Microwave needle operation .....	360
K.3. Acknowledgements .....	362
Appendix L: Thermophysical property estimation codes for steadily-propagating, gasless reacting materials.....	364
L.1. Running thermophysical property estimation code .....	364
L.2. Description of input and output .....	365
Input .....	366
Output .....	369
L.3. Variable description.....	369
L.4. Code structure.....	375
L.5. Summary.....	377
L.6. MATLAB code.....	377
Appendix M: Calibrating a spectrometer and 32-channel photomultiplier tube for temperature measurements.....	390
M.1. Procedure for calibrating camera pyrometer .....	390
M.2. Program for generating calibration file for 32-channel MC-PMT .....	394
Appendix N: High speed, 32-channel spectrometer temperature measurements in constant volume combustion cell.....	397
N.1. Running pressure cell temperature measurement code .....	397
N.2. Description of input and output .....	398
Input .....	399
Output .....	401
N.3. Variable description .....	401
N.4. Code structure .....	407
N.5. Summary .....	408
N.6. MATLAB code .....	408
Appendix O: MatVertilon – a multiplatform MATLAB program to extract data from Vertilon “.log” data files .....	420
O.1. Introduction .....	420
O.2. Running MatVertilon .....	420
O.3. MATLAB code .....	421
Bibliography .....	432
List of Publications .....	443

## List of Tables

Table 1-1: List of variables for Equations 1-9 through 1-13. ....	10
Table 3-1: Sample $\mu\text{Al}$ mass loadings as prepared (left) and after GO reduction (right). ....	73
Table 4-1: Summary of different length and time scales in this study. ....	96
Table 4-2: Sintering time, cooling time and size distribution of typical particles. Note: The cooling time for particle 1#, 2# and 4# is invalid because they were reheated by surrounding sintering particles. ....	106
Table 5-1: Mixture formulations of thermites added to PVDF/HPMC ink. Al wt.% includes the mass of $\text{Al}_2\text{O}_3$ shell. ....	113
Table 5-2: Oxygen release temperatures for different oxidizers as measured using temperature-jump time-of-flight mass spectrometry. ....	134
Table 5-3: Parameters for estimating heat transfer requirements in $\text{Al}/\text{I}_2\text{O}_5$ system for propagation. ....	134
Table 5-4: Measured and theoretical densities of the $\text{Al}/\text{I}_2\text{O}_5$ composites with calculated porosity values. ....	134
Table 5-5: Variables for equations 5-3 to 5-6. ....	136
Table 6-1: Mixture compositions for the two samples tested in this study. Number in sample description denotes atom ratio of reactants in composite. ....	142
Table 6-2: Calculated thermal diffusivities for tested SHS samples as estimated using single line temperatures and polynomial fits. Estimated kinetic parameters for tested SHS samples estimated using nonlinear regression fit and model described by Equation 6-8. ....	156
Table 7-1: Estimated thermal diffusivity of different materials used in this study. Data presented for mixtures were calculated on a mass-basis. ....	173
Table 8-1: Table of parameters. ....	199
Table 9-1: Energetic precursor formulations. ....	206
Table 9-2: Free-standing burn velocities in air and argon environments and ability to be ignited with microwaves. ....	220

Table 9-3: Physical constants used for the complex dielectric constant and complex magnetic constants used in calculations to estimate the intensity fraction absorbed and the maximum temperature. ....	227
--	-----

## List of Figures

Figure 1-1: (a) Wood burning as an example of a slow-burning fuel, burn time is on the order of minutes. (b) Firework with explosive charge at center as an example of explosives/pyrotechnics, burn time is on the order of microseconds ( $\mu\text{s}$ ). (c) Image of the Space Shuttle to demonstrate a typical propellant combustion, burn time on the order of milliseconds (ms). .....	2
Figure 1-2: Enthalpy of reaction on a volumetric and gravimetric basis of common monomolecular explosives and thermite mixtures. ....	16
Figure 1-3: Schematic showing the relative size, scales, and characteristics of homogeneous energetics, heterogeneous energetics, and metastable intermolecular composites for comparison purposes. ....	18
Figure 1-4: Data summary for burn time of metal nanoparticles and its dependency on particle size. Adapted from Chakraborty et al. <sup>58</sup> .....	20
Figure 1-5: (a) TEM image of as-received nanoaluminum fuel particles demonstrating the naturally passivating oxide shell. <sup>59</sup> (b) Plot of burning rate vs. fuel particle size. Adapted from Sullivan et al. <sup>38</sup> .....	21
Figure 1-6: Plot of $\text{O}_2$ release temperature in T-Jump/Time of Flight Mass Spectrometer vs. ignition temperature of the oxidizer when mixed stoichiometrically with aluminum nanoparticles. Adapted from Zarko et al. <sup>6</sup> .....	23
Figure 1-7: Depiction of (a) heterogeneous phase reactions vs. (b) condensed-phase interfacial reactions. ....	24
Figure 1-8: Images demonstrating reactive sintering which can negatively impact reactivity in energetic materials. (a) TEM image of a physically mixed nano Al/CuO energetic composite. (b) Image of the nano Al/CuO energetic composite after heating on a fast heating stage TEM. (c) Depiction of reactive sintering process. Adapted from Sullivan et al. <sup>47</sup> .....	26
Figure 2-1: Schematic of the experiment consisting of pressure cell and attached diagnostics. The pressure cell is shown on the left. The spectrometer is coupled to the pressure cell via a 1 m long optical fiber. The light from the fiber is spectrally dispersed by the selected grating on the turret which is subsequently imaged on the 32 channel PMT and digitized using the DAQ (Vertilon IQSP580). The digitized data is processed to produce time resolved spectra. ....	41
Figure 2-2: Temperature Measurement of Al/CuO nanothermite in pressure cell. a. Pressure-Normalized-Integrated Intensity, b. Temporal profile showing reaction temperatures. 1-2 Represents pressure drop; 2-3 Represents temperature drop and peak pressure; 3-4 Represents region of highest integrated intensity; 4-6 Represents	

region with sustained integrated intensity, increasing temperature, and declining pressure; 6-7 Represents temperature plateau and falling pressure. ....	48
Figure 2-3: Temporal Pressure-Temperature profiles for a. Al-Fe <sub>2</sub> O <sub>3</sub> , b. Al-Fe <sub>2</sub> O <sub>3</sub> -20WO <sub>3</sub> , c. Al-Fe <sub>2</sub> O <sub>3</sub> -60WO <sub>3</sub> , d. Al-Fe <sub>2</sub> O <sub>3</sub> -70WO <sub>3</sub> , e. Al-Fe <sub>2</sub> O <sub>3</sub> -90WO <sub>3</sub> , f. Al-WO <sub>3</sub> in pressure cell. The grey region is the error bound of the measurement. ....	51
Figure 2-4: Pressure-Temperature profiles for a. Effect of composition on temperature, b. Effect of composition on pressure and pressurization rate (Al/CuO: P <sub>max</sub> = 572 kPa and Press. Rate = 41 kPa/ $\mu$ s) and c. Effect of composition on Burn time. ....	53
Figure 2-5: High speed pyrometry frames of a. Al/Fe <sub>2</sub> O <sub>3</sub> , b. Al/Fe <sub>2</sub> O <sub>3</sub> /70% WO <sub>3</sub> and c. Al/WO <sub>3</sub> samples ignited on a hot wire at 1 atm. argon environment highlighting the gas production and enhanced reaction. In the Al/WO <sub>3</sub> case, the reaction propagated up the wire over a longer time scale due to lack of gas release from the oxidizer. In each figure, the top image is a gain-adjusted raw image and the bottom image is the 2-D temperature map. ....	55
Figure 2-6: Proposed reaction mechanism. ....	59
Figure 2-7: Calibration for the MC-PMT a. Evaluating the linearity of the detector by attenuating broadband signal using ND filters. Horizontal lines represent expected attenuation. b. Spectral response calibration using a high temperature Tungsten-Halogen lamp. ....	62
Figure 2-8: Spectral response curve of Vision Research Phantom Miro M110 high speed camera as obtained from the manufacturer. ....	63
Figure 2-9: Calibration for the Color camera pyrometer a. Spectral response calibration using a Black body and high temperature Tungsten-Halogen lamp. b. Evaluating the linearity of the camera sensor by attenuating broadband signal using ND filters. Dark line represents the expected attenuation. ....	64
Figure 2-10: Experimental for visualizing nanothermite reaction using high speed color camera. ....	65
Figure 2-11: Temperature measurement of Al/CuO nanothermite in pressure cell over extended durations measured without ND2 filter. The initial parts of the reaction is truncated due to saturated emission on the MC-PMT ....	65
Figure 2-12: Combustion of Al/Fe <sub>2</sub> O <sub>3</sub> observed using ND1 filter which is used for collecting early emission at the time of ignition. The gap in between is due to saturation of the sensor. ....	65
Figure 3-1: Apparatus mockup for thermal shock and temperature measurements of rGO/Al films. ....	70

Figure 3-2: SEM images of the original rGO/ $\mu$ Al sheets for mass loading ratios of (a) 4.2:1, (b) 2.1:1, and (c) 0.8:1. ....	73
Figure 3-3: High speed pyrometry frames of rGO/Al 2.1:1 sheet being thermally shocked with an 80 W CO <sub>2</sub> laser (90% power, 2.2 mm beam diameter) for 250ms in an Ar environment to produce nanoparticles. White areas represent pixels that were saturated. The top image is a gain-adjusted raw image and the bottom is the 2-D temperature map. Sample size is approximately 4.5 mm x 2.5 mm. ....	75
Figure 3-4: SEM images of rGO/Al films after thermal shock with a 32.7 W/mm CO <sub>2</sub> laser in an Ar environment. rGO/Al mass loadings/heating times are (a1) 4.2:1/100 ms (nanoparticles diffusing out of a shell), (a2) 4.2:1/250 ms, (a3) 4.2:1/500 ms, (b1) 2.1:1/100 ms, (b2) 2.1:1/250 ms, (b3) 2.1:1/500 ms, (c1) 0.8:1/100 ms, (c2) 0.8:1/250 ms, and (c3) 0.8:1/500 ms. For each experiment, the primary particle size was 3-4.5 $\mu$ m diameter Al. All images shown were taken from the first 1-2 layers below the surface of the rGO/Al sheet. ....	77
Figure 3-5: High speed pyrometry frames of rGO/Al 2.1:1 combusting in an O <sub>2</sub> flow (after a 250 ms thermal shock in Ar to generate nanoparticles). White areas represent pixels that were saturated and brown areas are high in error. The top image is a gain-adjusted raw image and the bottom is the 2-D temperature map. Sample size is approximately 4.5 mm x 2.5 mm. ....	79
Figure 3-6: (a) Average particle diameter of Al in rGO after thermal shock for different loadings. For data points not shown, particles were the ~3-4.5 $\mu$ m primary Al particles. (b) Burn time for 4.5 mm x 2.5 mm films versus laser-induced thermal shock time of the rGO/Al films at different loadings. ....	81
Figure 3-7: Images representing the Al particle and rGO sheet morphology as a function of position relative to the flame front during the combustion process. Degree of coalescence in Al in the products region demonstrates the impact of reactive sintering on the combustion process. Images were taken from a sample of rGO/Al 2.1:1. ....	82
Figure 3-8: TGA-DSC results of mass loss of GO during reduction. Sample was heated at 5°C /minute and held at 600°C for 120 minutes. Final mass was 42% of original. ....	83
Figure 3-9: Sample image of the exterior surface particles for a rGO/Al 4.2:1 film heated for 250 ms. The particles formed have larger particle sizes with a wider distribution in comparison to those found in the interior 1-2 layers (as shown in Figure 3-4). ....	84
Figure 3-10: TGA-DSC results of RGO heated in an O <sub>2</sub> environment. Sample was heated at 25°C /minute to 1200°C. Final mass was ~4% of original. ....	85

Figure 4-1: (a) Schematic showing high-speed microscope imaging of 3D-printed reactive materials. Read from right to left: (b) High-speed color video snapshot of reaction front; (c) Corresponding temperature map of reaction front; (d) Temperature map within a single particle, and (e) the same particle in SEM with its EDS map. CAD file of microscope objective courtesy of ThorLabs. .... 90

Figure 4-2: (a) Series of snapshots of burning flame front. Labelled on the left-top corner is the time after flame front entering the zone ( $512\text{ }\mu\text{m} \times 512\text{ }\mu\text{m}$ ). (b) The corresponding flame temperature maps and the average temperature (in Kelvin). High error points were marked as brown and are excluded from the calculations. (c) The mean (square marker) and median (round marker) flame temperature with time (with reference lines). (d) Schematic showing reactive sintering and its spatiotemporal relationship to reaction propagation. Note: the propagation direction is from right to left. .... 93

Figure 4-3: (a) Typical in-operando microscope image of Al/CuO reactive sintering, and after, (b) the corresponding SEM image of sintered combustion residue was found. (c) Series of snapshots of a single particle ( $55.55\text{ }\mu\text{s}$  per frame) showing the appearance of a large particle indicative of reactive sintering. (d) Intra-particle temperature at peak temperature showing a thermal gradient within particle. (e) The EDS mapping of the above in-operando sintered  $\text{Al}_2\text{O}_3$  particle coated with Cu nanoparticles, as evident by EDS mapping (red: Al; blue: O; green: Cu). Note: combustion is from right to left. .... 96

Figure 4-4: (a) Series of reactive sintering and ignition snapshots of a group of particles (labeled is frame sequence,  $55.55\text{ }\mu\text{s}$  per frame, scale bar:  $100\text{ }\mu\text{m}$ ) and (b) its corresponding schematic cartoon. Note: the marked arrows are propagating direction. (c) The corresponding SEM image of the sintered  $\text{Al}_2\text{O}_3$  particles coated with Cu nanoparticles, as evident by the EDS results in Figure 4-11 ..... 98

Figure 4-5: (a) Two typical temperature map snapshots of flame front of Al/CuO nanothermite and (b) its schematic showing of heat flux distribution from different stages of reactive sintering particles. .... 99

Figure 4-6: Tracking the time-resolved flame front position (propagation direction is from right to left). Left-top insert (a) shows, location where flame front is tracked and plotted and ordered 1-10, separated by  $50\text{ }\mu\text{m}$ . Insert b and c show two typical “reactive sintering” events. The “reactive sintering” are annotated with the dashed circles on the line plots. .... 100

Figure 4-7: Low (a) and high (b) SEM images, EDS results (c, Al; d, Cu) of a cross-sectional Al/CuO/HPMC/PVDF film. .... 104

Figure 4-8: The flame propagation observed in a  $512\text{ }\mu\text{m} \times 512\text{ }\mu\text{m}$  zone..... 105

Figure 4-9: Low (a) and high (b) resolution SEM images of the printed Al/CuO (90 wt.%) stick, and its burning snapshots in argon atmosphere (1 atm). .... 105

Figure 4-10: Low (a) and high (b) SEM images, EDS results (c and d) of a cross-sectional sintered particle. Based on a simplified estimation, one ~25-micron sintered $\text{Al}_2\text{O}_3$ , was formed by ~30 million ~80 nm Al NPs. The fine $\text{Al}_2\text{O}_3$ inside (~300 nm) was sintered by ~50 Al NPs. ....	106
Figure 4-11: Series of reactive sintering and ignition snapshots of a group particles (a, labeled is frame sequence, 55.55 $\mu\text{s}$ per frame). Note: the marked arrows are propagating direction. The corresponding SEM image (b) of the sintered $\text{Al}_2\text{O}_3$ particles coated with Cu nanoparticles, as evident by the EDS results (c). ....	106
Figure 5-1: Summary of combustion performance data from macroscale combustion experiments for different mixtures of $\text{Al}/\text{I}_2\text{O}_5/\text{CuO}$ . ....	118
Figure 5-2: (a) High-speed microscopy images for the $\text{Al}/75\% \text{I}_2\text{O}_5/25\% \text{CuO}$ sample and (b) corresponding temperature maps as calculated by color ratio pyrometry. (c) Drawing depicting different advection events that are observed in the reaction front for the high-speed microscopy videos. (d) Average particle velocity for particles that are ejected from the reaction front and example image for each mixture. Pictures in (d) have been brightened to clearly show streaks, but each picture has the same exposure time. ....	120
Figure 5-3: Post-reaction SEM images for different $\text{CuO}$ -doped $\text{Al}/\text{I}_2\text{O}_5$ composites and final particle diameter ( $D_f$ ). ....	123
Figure 5-4: Estimated values for the theoretical energy transfer and the amount of energy required to sustain combustion for different mixtures of $\text{Al}/\text{I}_2\text{O}_5/\text{CuO}$ . ....	129
Figure 5-5: Theoretical color channel ratios vs. temperature for the Phantom VEO710L and measured color ratios for the camera and lens assembly. ....	132
Figure 5-6: (a) Commercially purchased $\text{I}_2\text{O}_5$ crystals. (b1) SEM image of the $\text{Al}/\text{I}_2\text{O}_5$ high loading burn stick and (b2) corresponding element map for the sample. (c1) SEM Image of the $\text{Al}/75\% \text{I}_2\text{O}_5/25\% \text{CuO}$ high loading burn stick and (c2) corresponding element map for the sample. ....	132
Figure 5-7: Constant pressure (T HP) and constant volume (T UV) adiabatic flame temperature and equilibrium pressure for constant volume combustion. Constant volume calculations assume a 6% theoretical maximum density. ....	133
Figure 5-8: (a) High-speed microscopy images for the $\text{Al}/\text{I}_2\text{O}_5$ sample and (b) corresponding temperature maps as calculated by color ratio pyrometry. ....	133
Figure 5-9: (a) High-speed microscopy images for the $\text{Al}/95\% \text{I}_2\text{O}_5/\text{CuO}$ sample and (b) corresponding temperature maps as calculated by color ratio pyrometry. ....	133
Figure 5-10: Thermogravimetric analysis/differential scanning calorimetry results for $\text{I}_2\text{O}_5$ . ....	135



Figure 5-11: Estimated values for the theoretical energy transfer and the amount of energy required to sustain combustion for Al/CuO using constant pressure equilibrium calculations. Propagation velocity assumed to be 802 m/s. <sup>65</sup> .....	135
Figure 6-1: (a,b) Sample high-speed images of the propagating SHS samples tested. (c) Position vs. time data for combustion of 0.5 Al / 1.2 Zr / 1 C and 2 Al / 1.2 Zr / 1 C composites in Ar measured using high speed videography. Reported velocity is based on an average of linear fits between three experiments.....	150
Figure 6-2: (a,b) Sample color image and montage of temperature data collected via color ratio pyrometry. (c) Example of temperature vs. position data for both samples tested with regions of suspected phase transitions. Propagation direction is from left to right. Profiles shown are chosen for optimal comparison and not for the same timestamp. Dotted lines in the graph represent the light emission-based temperature measurements that are used to extend the dynamic range of the instrument.....	152
Figure 6-3: Temperature profile data for (a) 2 Al / 1.2 Zr / 1 C and (b) 0.5 Al / 1.2 Zr / 1 C printed samples. Solid line in each plot shows the goodness of fit for the exponential curve used to estimate thermal diffusivity on trailing edge. Shaded regions highlight the regions used for rise and decay time where temperature is <933 K.....	156
Figure 6-4: Plots of $d\eta/dt$ vs. time and corresponding Arrhenius model fit used to calculate the activation energy of (a) 2 Al / 1.2 Zr / 1 C and (b) 0.5 Al / 1.2 Zr / 1 C samples. Timestamps were normalized to time after the reaction front profile exceeded 933 K. ....	158
Figure 6-5: SEM visualization of reactant and product morphology of (a-b) 0.5Al:1.2Zr:1C and (c-d) 2Al:1.2Zr:1C stoichiometries. ....	160
Figure 6-6: (a) Depiction of COMSOL simulation to check influence of glass slide on temperature profile of the system. Figure includes glass slide. Initial temperature of hot region at t=0 is 1700 K. (b) Temperature profile in YZ plane for different times in simulations with and without glass slide. ....	160
Figure 6-7: Plot of normalized light intensity vs. time for the 2 Al / 1.2 Zr / 1 C composite to demonstrate the periodic oscillations in brightness during reaction propagation. ....	161
Figure 6-8: (a) High-speed microscope images of propagation for the 0.5 Al / 1.2 Zr / 1 C samples and (b) corresponding temperature map as measured by color pyrometry. Images captured using a Phantom VEO710L high speed camera coupled to an InfinityOptics K2 DistaMax Lens with a resolution of ~1.7 $\mu\text{m}/\text{px}$ .....	161
Figure 7-1: Macro-scale burn rate images of a printed film on a microscope slide at 5,000 frames/s. Burn time measured from ignition to the time that the reaction front reaches the end of the microscope slide. Note: Propagation is from left-to-right. ...	166

Figure 7-2: High-speed microscopy images for (a) Al/PVDF/C and (b) Al/PVDF/SiO <sub>2</sub> with outlines of reaction front. Red lines represent regions that had not moved within 2 frames. (c) Plots of window-size normalized flame length as a function of time for Al/PVDF/C and Al/PVDF/SiO <sub>2</sub> . (d) Average flame length and velocity for additive-doped films. Note: Propagation is from left-to-right. ....	168
Figure 7-3: (a) Al/PVDF/C and (c) Al/PVDF/SiO <sub>2</sub> imaged through high-speed microscopy with area outline (red/white) and velocity vectors (blue). Histogram of measured velocities for (b) Al/PVDF/C and (d) Al/PVDF/SiO <sub>2</sub> . ....	168
Figure 7-4: (Top) High-speed microscopy images and (bottom) corresponding temperature maps obtained by color camera pyrometry.....	169
Figure 7-5: Scanning electron microscopy images and elemental maps of typical cross-sections for Al/PVDF/C film (left) and Al/PVDF/SiO <sub>2</sub> (right) films showing distributions of large particle additives in each film.....	172
Figure 7-6: Images of the Al/PVDF/C (left) and Al/PVDF/SiO <sub>2</sub> (right) films prior to ignition as observed against the microscope slide while focusing for the high-speed microscope. ....	173
Figure 8-1: Spectrum of thermal conductivities typically used in polymer composites with tunable thermal properties. ....	176
Figure 8-2: Relative thermal diffusivity ( $\alpha f/ar$ ) mapping with respect to volume fraction for (a) simple homogenously mixed, uniform distribution, and (b) distribution of rod of variable aspect ratios as well as an example of (c) the heterogeneity scaling of solid energetic constituents relative to $\delta$ . ....	181
Figure 8-3: (a) Flame front snapshot and (b) time resolved behavior of the accumulated area of the pre-heated zone, total burned material, and burned energetic material for the case of AR=55, Vol.%=20, t=480 $\mu$ s (left to right propagation). ...	185
Figure 8-4: Effect of aspect ratio and vol.% of rods with a fixed thermal conductivity on (a) the change in total area burned, (b) energy released over the allotted burn time, and (c) the extent of pre-heating during flame propagation. ....	186
Figure 8-5: Effect of thermal conductivity and vol.% of rods with an aspect ratio (AR=45) on (a) the change in total area burned, (b) energy released over the allotted burn time, and (c) the extent of pre-heating during flame propagation. ....	190
Figure 8-6: Thermal mapping of energetic material propagation integrating rods with varied thermal conductivities within an invariant AR=45 rod mapping of 20 vol.% at t=325 $\mu$ s (Propagation from left to right).....	191

Figure 8-7: (a) Grid of relevant temperature nodes for position index $p,q$ at the $(n+1)^{\text{th}}$ time step with (b) visualization of the implicit finite difference method using adjacent temperature nodes in the previous time step.....	194
Figure 8-8: Thermal resistance approach for defining effective thermal conductivity at the boundary of discrete thermal conductivity distribution. ....	196
Figure 8-9: Thermal mapping of energetic material propagation incorporating rods of constant thermal conductivity ( $k=k_0 10^4$ ) with varied aspect ratios at 30 vol.% at $t=325 \mu\text{s}$ (Propagation from left to right). ....	198
Figure 8-10: Truncated fluctuations of the extent of pre-heating as a function of time during flame propagation. ....	199
Figure 9-1: (a) Needle-based microwave ignition experiment at 2.45 GHz. Peak power of the entire system is $\sim 15\text{W}$ . (b) Simulations of the electric field at the tip of the needle for a frequency of 2.45 GHz (using Ansys HFSS and an auto-generated mesh).....	209
Figure 9-2: (a) Calculated fraction of microwave (2.45 GHz) intensity absorbed for bare and core-shell Al and Ti nanoparticles in a $25 \mu\text{m}$ diameter cylinder as a function of depth and (b) the estimated maximum temperature after being exposed to a microwave input of 15 W for 0.5 s. Physical properties used in calculation are provided in Table S1. ....	213
Figure 9-3: (a) Ignition of nano-Ti powder in air (50nm diameter) using microwave needle apparatus imaged using a high-speed camera. $t=0$ defined as starting time for power supplied to the applicator. Dashed white line indicates position of needle. (b) Simulation of the experiments in (a) showing the near field electric field on the ground plane performed using Ansys HFSS and an auto-generated mesh. ....	216
Figure 9-4: Ignition of Ti/PVDF (65 wt.% Ti) film in air using microwave needle apparatus imaged using a high-speed camera. $t=0$ defined as starting time for power supplied to the applicator. Dashed white line indicates position of needle. ....	219
Figure 9-5: (a) Depiction of the architecture for a layered Al/PVDF, Ti/PVDF propellant that can be initiated with microwaves where layers 1-4 are Al/PVDF ( $\Phi=1$ ) and layer 5 is Ti/PVDF (65 wt.% Ti) ( $\sim 35 \mu\text{m}$ thick). (b) SEM/EDS image of layered film cross-section as outlined in a. (c) EDS line scan results for line drawn in b. ...	222
Figure 9-6: Microwave needle ignition of a layered Al/PVDF, Ti/PVDF propellant where layers 1-4 are Al/PVDF ( $\Phi=1$ ) and layer 5 is Ti/PVDF (65 wt.% Ti). (a) Al/PVDF region does not ignite when exposed to microwave radiation and (b) Ti/PVDF layered region ignites and propagates into Al/PVDF region. ....	223

Figure 9-7: Transmission electron microscopy images of the as-received (a) nano-Aluminum (Novacentrix) and (b) nano-Titanium (US Research Materials). Images taken with a FEI Tecnai12. ....	226
Figure 9-8: Scanning electron microscopy images of the cross-section for as-received (a) 3-4.5 $\mu$ m diameter Al micron particles (Alfa Aesar) and (b) 10-14 $\mu$ m diameter Al micron particles (Alfa Aesar). Images taken with an FEI NNS450. ....	226
Figure 9-9: Scanning electron microscopy images of the cross-section for (a-b) Al/PVDF and (c-d) Ti/PVDF films. Each printed sample is 4 layers thick. Images taken with FEI NNS450. ....	227
Figure 9-10: Macroscopic burn test apparatus schematic with gas inlets allow for environment control and ignition of a free-standing film using a resistively heated nichrome wire. ....	227
Figure 9-11: Images from microwave heating and ignition experiments for nano-titanium as performed in a glass tube with an inert Argon environment. Right image shows small heating point immediately around the microwave source. ....	228
Figure 9-12: Distance vs. time data used to estimate propagation rate of a Al(25)/PVDF(75) mixture in air and argon. ....	228
Figure 9-13: Distance vs. time data used to estimate propagation rate of a Al(65)/PVDF(35) mixture in air and argon. ....	229
Figure 9-14: Distance vs. time data used to estimate propagation rate of a Ti(35)/PVDF(65) mixture in air. This sample did not propagate in argon. ....	229
Figure 9-15: Distance vs. time data used to estimate propagation rate of a Ti(65)/PVDF(35) mixture in air and argon. ....	230
Figure 9-16: Ti-PVDF argon environment post combustion products. ....	230

## List of Abbreviations

$A$	Pre-exponential factor for reaction rate Burning surface area
$A_E$	Cross-sectional burned area
$A_{tube}$	Cross-sectional area of tube (m <sup>2</sup> )
AR	Aspect ratio
bpp	Bits per pixel
$c$	Speed of light
CEA	NASA “Chemical Equilibrium with Applications” program
$C_i$	Concentration of reactant “i”
$C_{ij}$	Calibration factor of channel
CHNO	Carbon, hydrogen, nitrogen oxygen
CL-20	Hexanitrohexaazaisowurtzitane (monomolecular explosive)
CMOS	Complementary metal oxide semiconductor
$C_{1,2}$	First and second radiation constants
$C_{p,i}$	Constant pressure heat capacity of species “i”
$d$	Particle diameter
$D$	Mass diffusivity Diameter
DAQ	Data acquisition unit
DMF	N,N-dimethylformamide
$D_r$	Aggregate projected area
DSC	Differential scanning calorimetry
$E$	Electric field
$E_a$	Activation energy

EDS, EDX	Energy dispersive x-ray spectrometry
$E_p$	Energy carried by a single particle
FWHM	Full width half-max
GO	Graphene oxide
$h$	Planck's constant Convective heat transfer coefficient
$H$	Magnetic field
$\bar{h}_i$	Specific enthalpy of species "i"
$\bar{h}_{f,i}^o$	Formation enthalpy at a reference temperature and pressure
HMX	1,3,5,7-Tetranitro-1,3,5,7-tetrazoctane (monomolecular explosive)
HPMC	Hydroxypropyl methylcellulose
HTPB	Hydroxyl-terminated polybutadiene
$H_{react}$	Enthalpy of reaction
$\Delta H_{vap}$	Heat of vaporization
$I_0$	Incident energy
$I_{abs}$	Incident energy absorbed
$I_i$	Intensity of channel "i"
$k$	Thermal conductivity
$K$	Boltzmann constant Temperature-based kinetic term
$k$	Thermal conductivity
$k_{eff}$	Effective thermal conductivity
$k_n$	Reaction rate constant of reaction "n"
$L$	Sample length (m) Radiance
LED	Light emitting diode

MC-PMT	Multi-channel photomultiplier tube
$m_i$	average molar mass of particle I (kg/mol)
MIC	metastable intermolecular composite
MS	Mass spectrometry
MW	Microwave
$n$	Reaction order
$N$	Number of particles
$N_i$	Number of moles of species “i”
$N_p$	Number of advected particles per unit length (#/m)
NPs	nanoparticles
$P$	Pressure Power
$P_d$	Power dissipated
PETN	2,2-Bis[(nitrooxy)methyl]propane-1,3-diyl dinitrate (monomolecular explosive)
PMT	Photomultiplier tube
PTFE	Polytetrafluoroethylene
PVDF	Polyvinylidene fluoride
$Q_f/Q_{tot}$	Ratio of forward mass transfer to total transfer
R	Gas constant
RDX	1,3,5-Trinitro-1,3,5-triazinane (monomolecular explosive)
RGB	Red, green, blue
$Q$	Internal energy generation rate
rGO	Reduced graphene oxide
SEM	Scanning electron microscope
$Sh$	Sherwood number

SHS	Self-propagating high-temperature synthesis
$t$	Time
$t_d$	Decay time
$t_r$	Rise time
$T$	Temperature
$T_0, T_a$	Initial temperature
$T_{ad}$	Adiabatic flame temperature
$t_{burn}$	Burn time
TEM	Transmission electron microscope
TGA	Thermogravimetric analysis
$T_{ign}$	Ignition temperature
T-Jump	Temperature jump
TNT	2,4,6-trinitrotoluene (monomolecular explosive)
TOFMS	Time of flight mass spectrometry
$v$	Propagation velocity
$V$	Volume
$V_p$	Volume of particle "i" ( $m^3$ )
$x$	Position
$\Delta x$	Distance
XRD	X-ray diffraction
$y$	Position
$\alpha$	Thermal diffusivity
$\beta$	Oxidizer mass flux
$\delta$	Characteristic length



$\varepsilon$	Emissivity Permittivity
$\eta$	Conversion
$\lambda$	Wavelength
$\mu$	Permeability
$\rho$	Density
$\rho_i$	Density of particle “i” (kg/m <sup>3</sup> )
$\sigma$	Stefan-Boltzmann constant Electrical conductivity
$\Phi$	Porosity of sample Equivalence ratio Conversion-based kinetic term
$\tau$	Characteristic time
$\tau_{diff}$	Diffusion time
$\tau_{kin}$	Kinetic reaction time
$\tau_{rxn}$	Characteristic reaction time
$\chi_c$	Dimensionless length
$\chi_i$	Spectral response as a function of wavelength
$\psi_i$	Channel gain of channel “i”
$\dot{\omega}$	Reaction rate

# Chapter 1: Introduction

## *1.1 Combustion and energetic Materials*

Every material has energy stored in the chemical bonds between the atoms that make it up. Chemical reactions can weaken or break these bonds to convert the stored potential energy to kinetic energy in a variety of forms including light, heat, and pressure. For those that are trying to capitalize on this stored potential energy, it would be desirable that the reactions are exothermic; in other words, the reaction that the material undergoes releases more energy than required to make the reactants or start the reaction. The most common exothermic reaction is combustion, a simple process whereby a fuel and oxidizer react and release energy (typically in the form of heat). It would be difficult to overstate the importance of combustion in history or our daily lives; it has been used as the primary energy generation method since the discovery of fire and has enabled man to cook foods, power homes, create new materials, defend itself, and explore the universe.

Almost all materials burn, though how the materials burn can vary widely between compounds. The amount of energy stored, how much energy is required to start the combustion process, the chemical makeup of the final products, and the chemical reaction rate can take place all factor into the combustion characteristics of a material. In most materials that could ever be used as fuels, the range of times to release all of the stored energy extends far beyond many minutes or hours; however, some select compounds that release this energy seemingly instantaneously on time scales ranging from micro- to milliseconds (see Figure 1-1). These compounds are commonly

referred to as “energetic materials” and their reactions are generally only limited by how fast the fuel and oxidizer can reach each other. The mode of energy release in energetic materials is typically characterized by large heat and pressure waves following the reaction due to the ultrafast nature of their combustion.<sup>1</sup>



**Figure 1-1:** (a) Wood burning as an example of a slow-burning fuel, burn time is on the order of minutes. (b) Firework with explosive charge at center as an example of explosives/pyrotechnics, burn time is on the order of microseconds ( $\mu\text{s}$ ). (c) Image of the Space Shuttle to demonstrate a typical propellant combustion, burn time on the order of milliseconds (ms).

Energetic materials can be further subcategorized into propellants (e.g. solid rocket motors), pyrotechnics (e.g. fireworks), and explosives (e.g. TNT), each with slight variations in combustion time and typical purpose. The introduction to this dissertation provides a general overview of combustion in all types of energetics, but the research efforts herein primarily revolve around propellants and pyrotechnics.

### 1.1.1 History of energetic materials

An extensive amount of research has been dedicated to refining the synthesis of energetic materials, characterizing their combustion, and improving performance (likely since the second they were discovered). Their history is only briefly described here, but the very existence of energetic materials has shaped the world as we know it. Those interested in a very detailed history should refer to the U.S. Department of the Army Handbook on Military Explosives.<sup>2</sup> Energetic materials have been reported as

military devices as early as ~424 B.C. and have since led to advancements in propellants, pyrotechnics, and explosives through the efficient storage and release of the energy within their atomic bonds.<sup>2</sup>

Historically, a majority of energetic materials have been developed as part of improvements in weapons whether it be in an act of aggression or for self-defense. As described in Denny & McFadzean's *Rocket Science*, "One uniquely human characteristic is our proclivity and ability to design and build machines for the sole purpose of beating the crap out of other members of our own species."<sup>3</sup> Flames and incendiaries (which are material that readily ignite and quickly spread flames) date back to the age of the caveman as methods to ward off enemies and were even employed during biblical war times.<sup>2</sup> Historical development of incendiaries has led to the invention of early flamethrowers, fire-arrows, and Greek Fire, though these were primitive energetic materials to those being explored in the East.<sup>2</sup> By 1249, the Chinese had developed pyrotechnics for fireworks and eastern countries had started experimenting with early iterations of gunpowder explosives.<sup>2</sup> Experimentation on the recipe for and manufacture of gunpowder with different proportions of saltpeter, charcoal, and sulfur were carried out by the Europeans and revolutionized the battlefield alongside the invention of the gun, cannon, and missile.<sup>2</sup> Perhaps one of the most important discoveries which came in tandem with the general recipe for gunpowder was the recognition that the combustion performance of these materials could be easily altered by changing the gunpowder particle size and the packing density in the charge.<sup>2,4</sup> Gunpowder formulations with different ratios, additives, and

geometries saw use in both military and civilian applications (land forming, mining, etc.) and are still commonly used today.

However, modern energetic materials truly began taking shape in the 19<sup>th</sup> century. Molecular explosives like ammonium nitrate ( $\text{NH}_4\text{NO}_3$ ), ammonium perchlorate ( $\text{NH}_4\text{KClO}_4$ ), trinitrotoluene (TNT,  $\text{C}_6\text{H}_2(\text{NO}_2)_3\text{CH}_3$ ), nitroglycerin ( $\text{C}_3\text{H}_5(\text{NO}_3)_3$ ), and so many more were all developed and saw their first uses all within a ~60-year period spanning from 1800-1860. Similar to the widespread employment of gunpowder, molecular/organic explosives saw use as part of armaments, munitions, mining operations, land forming, and more (oddly enough, nitroglycerin is also a very popular medicine). The development of energetic materials became so lucrative that some prominent pioneers left behind massive corporations and endowments (such as Alfred Nobel, the inventor of dynamite and the associated Nobel Prizes).

The next era of major advancements in energetic materials and their uses came during World War I and World War II. Incendiaries, pyrotechnics, explosives were used heavily on the battlefield for everything from munitions to lighting with new methods of activation, adoption of large-scale processing for the war effort, and new formulations to enhance performance (such as the introduction of aluminum/metal additives).<sup>2,4</sup> This period of time also saw the first long-range rockets using both solid and liquid energetic materials as propellants with pioneers like Robert H. Goddard and Wernher von Braun on either side of the conflict.<sup>2,3,5</sup>

Although many efforts from this point forward were still being made to forge new and exciting ways to wreak havoc (such as shape charges, plastic explosives, nuclear weapons, etc.), it also was the origin of modern rocketry.<sup>3,5</sup> The technology that

was born out of the development of the ballistic missile era also gave birth to the development of the Redstone rocket, Saturn V, and Space Shuttle, ultimately enabling humans to explore the stars. Rocket propulsion is not a new field by any stretch of the imagination – primitive rockets were reported to exist alongside fireworks in China – though it is imperative that we continue to invest time and effort into the formulation, refinement, and development of new energetic materials with hopes of discovering new ways to generate energy and further our reach into the universe.

Solid propellants (like those that will be explored in this dissertation) are more akin to gunpowder than TNT in terms of reactivity and have unique advantages over liquid energetics (storability, handling, etc.).<sup>3,5</sup> However, they have seen only limited use in the past few decades because they are just as uncontrollable as both gunpowder and TNT. Once a solid rocket motor is ignited, it is going to burn until there is either nothing left to burn or there is a catastrophic failure (though it might still burn after the failure, just in smaller pieces). This is in stark contrast to the simplicity of a liquid rocket engine which has a throttle and can be stopped/restarted. Extensive research efforts that have been put forth in both the military and civilian communities are attempting to spur technology advancements in solid propellants with enhanced safety, efficiency, and versatility. In this dissertation, new ways to synthesize and characterize energetic materials with specific applications to propulsion are explored.

#### 1.1.2 Chemical reaction kinetics and transport phenomena basics

Chemical reactions can be generally represented by change of a chemical species – a simple example of these reactions can be seen in Equation 1-1 where a reactant (A) is reacting to form a final product (C) with an intermediate product (B).

However, in the grand context of chemical engineering and reaction dynamics, time plays a critical role to evaluate reaction kinetics. To estimate the formation rates of chemicals A, B, and C, the system of differential equations shown in Equations 1-2, 1-3, and 1-4, respectively, can be solved where  $C_i$  is the concentration of chemical “i,”  $k_n$  is the reaction rate constant for reaction “n,” and  $t$  is time. Now, these equations could be modified to represent a more real system with reversible reactions, equilibrium constant expressions, temperature-dependent reaction constants, etc., but the reaction set below is the most simplified case. The solution to this set of differential equations is relatively straightforward, but there are numerous terms in the final solution for  $C_c$  and adding more detailed kinetics terms would only further complicate the result.



$$\frac{dC_A}{dt} = -k_1 C_A \quad (1 - 2)$$

$$\frac{dC_B}{dt} = k_1 C_A - k_2 C_B \quad (1 - 3)$$

$$\frac{dC_C}{dt} = k_2 C_B \quad (1 - 4)$$

Combustion can be generally modeled the same way, however, the calculations become infinitely more complex when considering all of the possible intermediates that can occur at the elevated temperatures.<sup>1</sup> As an example, the water formation reaction ( $H_2 + \frac{1}{2} O_2 \rightarrow H_2O$ ) has as many as 40 chemical reactions involved in the mechanism, all of which have their own reaction constants, equilibrium expressions, and temperature dependencies and considers at least 8 species (even though there are only 3 in the overall expression).<sup>1</sup> This might be the most simplistic combustion reaction, so it can only be imagined the number of chemical reactions must be considered for

hydrocarbon or metallic combustion when different phases of matter and a plethora of atoms must be considered; the number of reactions and chemicals can extend into the thousands (at least).

It is commonly said that chemical reactions are “only as fast as the slowest step,” so to simplify reaction kinetic analysis for energetic materials it only makes sense to consider the “slowest step.” Combustion is a very fast process though, and these reaction kinetic constants are typically so fast that they are not actually the rate limiting step – diffusion is. Proximity of the reactants to each other is essential to initiation and sustained reaction, but in combustion, the diffusion of the fuel and oxidizer can be rate-limiting due to the heightened reaction rates.<sup>1</sup> In the case of homogeneous (or monomolecular) energetic materials such as TNT, RDX, and HMX, the diffusion distance for the fuel and oxidizer is on the angstrom scale ( $10^{-10}$  m) since they are different functional groups on the same molecule.<sup>6,7</sup> Diffusion rates can be modeled using Fickian diffusion with careful consideration of the diffusion coefficient dependency on pressure and temperature.<sup>1,8</sup> Considering that the primary mechanism of mass transfer between the fuel and the oxidizer is diffusion, the characteristic diffusion time ( $\tau_{diff}$ ) for the mixture scales roughly as shown in the Equation 1-5 below where P is pressure, T is temperature, and  $\Delta x$  is the distance.<sup>1,8</sup> Therefore, in conditions where the pressure and temperature are to remain constant, it can be said that the initiation step scales proportionally with the distance separating the fuels and the oxidizers. This is an important concept that will come up in the discussion comparing homogeneous vs. heterogeneous energetic composites.

$$\tau_{diff} \propto \frac{P\Delta x}{T^{1.5}} \quad (1 - 5)$$



Now, the term “energetic materials” implies that some amount of energy is released from the materials during the reaction. The liberated energy typically comes in the form of heat and the amount of energy actually released can be computed by comparing the enthalpies of the reactants and products of the reaction as shown in Equation 1-6 where  $H$  is enthalpy,  $N_i$  is the amount of species “i”, and  $\bar{h}_i$  is the specific enthalpy of species “i.”<sup>1</sup> The specific enthalpy of the species can be calculated using Equation 1-7 where  $\bar{h}_{f,i}^o$  is the formation enthalpy at a reference temperature and pressure (typically 298 K, 1 atm). By combining Equations 1-6 and 1-7, the adiabatic flame temperature ( $T_{ad}$ ) can be computed. The adiabatic flame temperature represents the equilibrium temperature that would be achieved for a given set of reactants and products and represents the “best case scenario” flame temperature. The calculation of adiabatic flame temperature is also dependent on whether the combustion is to occur under constant-pressure or constant-volume conditions (the latter having a higher  $T_{ad}$ ). Calculating adiabatic flame temperatures is an art to some extent since it requires detailed knowledge on the thermophysical properties of the reactants and predicted products to get accurate results, but considerable effort has been made in programs like CHEETAH (Lawrence Livermore National Laboratory) and CEA (NASA).<sup>9,10</sup>

$$H_{react} = \sum_{react} N_i \bar{h}_i = H_{prod} = \sum_{prod} N_i \bar{h}_i \quad (1 - 6)$$

$$\bar{h}_i = \bar{h}_{f,i}^o + \int_{298}^T \bar{C}_{p,i} dT \quad (1 - 7)$$

In the grander scheme of propellants and propagating reactions, both temperature and propagation velocity are important in determination for performance

and mission suitability. For example, combustion temperature and mass burning rate are both critical parameters in designing a solid rocket motor as it will affect parameters like thrust profiles, nozzle diameter, exit area, and will even play a role in material selection for the construction of the rocket.

Propagation rates in diffusion-limited reactions can be distilled down to the expression shown in Equation 1-8 where  $\alpha$  is thermal diffusivity of the composite and  $\dot{\omega}$  is the reaction rate for the system. Simply speaking, the propagation velocity is dependent upon how fast the energy can be released (represented by  $\dot{\omega}$ ) and how fast that energy can be transferred to the next unreacted segment of material (represented by  $\alpha$ ). As can be seen in the expression, propagation velocity is also inherently entangled with the combustion temperature since it is dependent upon the reaction rate of the process. This expression for propagation is often seen applied to gaseous diffusion flames, but it can be reasonably conjectured that it would apply to condensed phase reactions.

$$v \propto \sqrt{\alpha \cdot \dot{\omega}} \quad (1 - 8)$$

In an ideal case, it would be desirable for the reaction front to actually sustain propagation using only the energy liberated by the chemical reactions. To sustain propagation, the heat released by the reactions needs to meet or exceed the energy required to raise the temperature of the unreacted material to the ignition temperature ( $T_{\text{ign}}$ ). Egan et al. did a simple analysis in their article where the amount of energy transferred per unit length of material reacting was estimated for the different modes of heat transfer: conduction, radiation, convection, vaporization/condensation, and advection.<sup>11</sup> A more refined estimation model for energy transfer per unit length of

material reacting in a theoretical burn tube are represented by Equations 1-10 through Equation 1-14 (table of variable descriptions provided in Table 1-1) for conduction, radiation, convection, vaporization, and advection, respectively. This generalized heat transfer estimate model can be used roughly to assess the likelihood of a given sample propagating or to diagnose any problems that may complicate propagation (as will be explored in this dissertation).

$$\frac{Q_f}{Q_{tot}} = \frac{\Phi}{1 + \Phi} \quad (1 - 9)$$

$$E_{cond} = \frac{kA_{tube}(1 - \Phi)}{v_{prop}} \left( \frac{dT}{dx} \right) \quad (1 - 10)$$

$$E_{rad} = \frac{\sigma \varepsilon (1 - \Phi) A_{tube} (T_{ad} - T_0)}{v_{prop}} \quad (1 - 11)$$

$$E_{conv} = \left( \frac{Q_f}{Q_{tot}} \right) \left( \frac{1}{L} \right) \sum_i N_{i,gas} \left( \bar{h}_i(T_{ad}) - \bar{h}_i(T_{ign}) \right) \quad (1 - 12)$$

$$E_{vap} = \left( \frac{Q_f}{Q_{tot}} \right) \left( \frac{1}{L} \right) \sum_i N_{i,vap} \left( \bar{h}_i(T_{ad}) - \bar{h}_i(T_{ign}) \right) \quad (1 - 13)$$

$$E_{adv} = \sum_i \left( \frac{\rho_i V_i}{m_i} \right) \left( \bar{h}_i(T_{ad}) - \bar{h}_i(T_{ign}) \right) \quad (1 - 14)$$

**Table 1-1:** List of variables for Equations 1-9 through 1-13.

Symbol	Name
$k$	Thermal conductivity (W/mK)
$\sigma$	Stefan-Boltzmann constant (W/m <sup>2</sup> K <sup>4</sup> )
$\varepsilon$	Emissivity
$T_{ad}$	Adiabatic flame temperature (K)
$T_{ign}$	Ignition temperature (K)
$T_0$	Initial temperature
$dT/dx$	Thermal gradient (K/m)
$Q_f/Q_{tot}$	Ratio of forward mass transfer to total transfer
$\Phi$	Porosity of sample
$v_{prop}$	propagation velocity (m/s)
$A_{tube}$	Cross-sectional area of tube (m <sup>2</sup> )

$L$	Sample length (m)
$N_p$	Number of advected particles per unit length (#/m)
$\rho_i$	Density of particle $i$ (kg/m <sup>3</sup> )
$V_p$	Volume of particle $I$ (m <sup>3</sup> )
$m_i$	average molar mass of particle $I$ (kg/mol)

---

Ultimately, this abbreviated introduction to reaction kinetics and transport phenomena relevant to combustion should demonstrate that combustion is a complex amalgam of heat and mass transfer with integrated chemical reaction kinetics. Each segment of the concepts described above can be further dissected when considering different additives, material geometries, environments, etc. Perhaps more impressively, all of these processes happen in the blink of an eye. Structure-function relationships will be explored in this dissertation using reaction kinetics and transport phenomena as guiding principles to understand changes in reactivity and propagation of energetic materials.

### *1.1.3 Combustion performance characterization techniques*

Characterization of combustion performance in energetic materials is paramount to optimizing formulations, tailoring performance, and designating applications for materials. While the amount of energy theoretically released and transferred during a reaction is readily calculated, actual experimental measurements are important for testing models under real world conditions. Considering the large body of work that has been dedicated to identifying new characterization techniques both prior to and during the research that was carried out over the period of time this dissertation was performed, this section will be intentionally kept brief to focus on key experiment descriptions and major recent developments that are relevant.

#### *1.1.3.1 Peak pressure and pressurization rate measurements*

As previously discussed, combustion is primarily used for energy release with the intention of performing some work on a system. This work is typically accomplished with expansion of gases at high temperatures that drives a mechanical system. Pressure is the most common metric to be measured as potential to do work on a system for any given mixture since it is the amount of force which can be exerted over a (presumably) known area, and pressure profiles can provide information about the impulse or total burn time. In the context of propulsion, the pressure and pressurization rate are critical parameters for determining effectiveness of a propellant mixture and designing the propulsion system itself.<sup>5</sup> Constant volume combustion cells equipped with pressure transducers have been widely used to estimate pressurization rates in energetic materials in confined environments, while constant pressure experiments may be conducted using simple cantilever beam setups.<sup>6</sup> Though, it is important to note that there is a critical difference between constant volume and constant pressure characterization results since some energy is lost to the expansion of gas in the constant pressure case (i.e. peak temperatures and pressures are lower in isobaric conditions).

#### *1.1.3.2 Temperature measurements*

Measuring temperature is another important metric used to characterize energy release in combusting systems. However, actually measuring the temperature of a reacting system can be deceptively difficult and multiple methods have been developed to probe temperature dynamics. Most temperature measurement methods can be classified as either invasive (implementation may skew results) or noninvasive

(measured remotely). Thermocouples are the most widely used “invasive” temperature measurement methodology, however they offer relatively slow response times, are highly dependent on placement in the material, and can even act as heat sinks on smaller scale experiments. Non-invasive techniques such as light emission measurements are more commonly employed as temperature measurement methods for energetic material combustion since setup is relatively simple, data can be collected from a distance without directly influencing the combusting material, and modern optical equipment has good light sensitivity to accommodate the high data acquisition rates.

In the simplest form, light emission can be measured by directing light into an optical apparatus that can split light into multiple bands at different wavelengths of light which can all be independently measured. A key aspect of this design is having high light sensitivity so that light emission can be recorded as a function of time at high data acquisition rates. Recent advances in data acquisition and optical equipment has enabled light emission measurements to be performed at >100 kHz where temperature can be estimated by assuming that the combusting body emits like a blackbody or graybody (or by applying some other wavelength dependent emission assumption).<sup>12,13</sup> Peuker et. al also discuss possible modifications that can be made to spectrometry apparatuses which focus specifically for light emission of intermediate products (notably AIO) that can be used to extrapolate reaction proceedings and temperature.<sup>14</sup>

However, while these spectrometry-based measurements offer high data acquisition rates and sensitivity, they fail to capture the spatial dynamics of reactions. As a solution to this, Densmore et. al demonstrated that a high-speed color camera could be used as an effective temperature measurement tool which eliminates

dependence on experimental variables (e.g. viewing angle) by focusing on ratios in light intensity for different color channels.<sup>15</sup> In a separate article, Kim et. al demonstrated that light intensity alone could even be reliably used as a temperature measurement method using grayscale images.<sup>16</sup> Recently experiments by Alemohammad et. al have also demonstrated the use of hyperspectral imaging to record the spatial dynamics of a reaction with the capability to measure light emission at different positions in a single image.<sup>17</sup> Unfortunately, many of the non-invasive techniques, especially those that cannot record emission as a function of wavelength, are subject to error based on emission properties of different materials at high temperature.<sup>14,18–20</sup>

Ultimately, all of these methods are viable options to measure temperature of a reacting system with uniquely different advantages depending on the application. This dissertation discusses the development of a high-speed color ratio pyrometry apparatus based on the work by Densmore et. al<sup>15</sup> in Chapter 2.

#### 1.1.4 Homogeneous vs. heterogeneous energetic materials

At this point, it is relevant to talk about the different types of energetic materials. Traditionally, energetic materials have been categorized into two distinct classes – heterogeneous and homogeneous energetic composites.<sup>2,6,7,21–23</sup>

Homogeneous energetic materials are monomolecular compounds that combine the fuel and oxidizers of the combustion reaction on the same molecule in the form of different functional groups.<sup>2,6,7,21,23</sup> The most common homogeneous energetics have characteristically organic features including abundant amounts of carbon-, oxygen-, and nitrogen-based functional groups.<sup>2</sup> Homogeneous energetic materials are usually

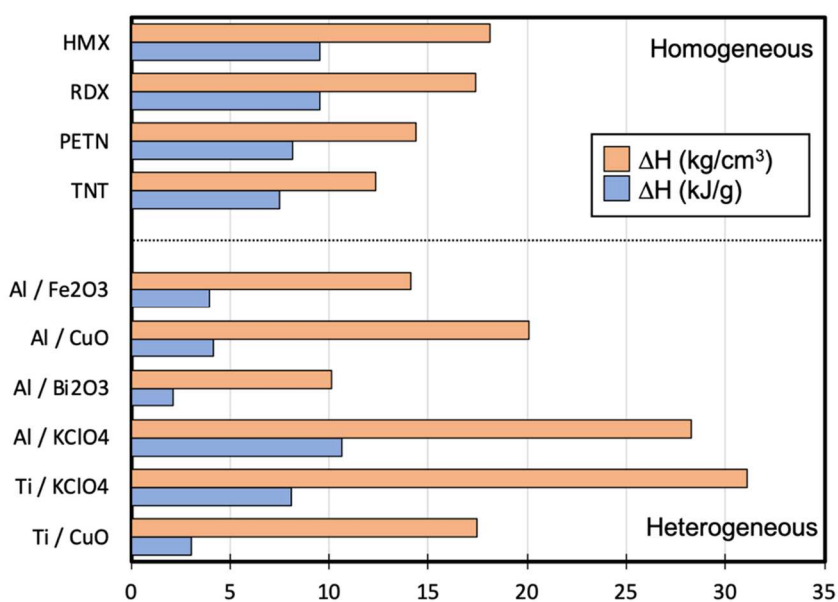
the first things that come to mind when people think “explosives” like TNT, RDX, HMX, PETN, etc. These organic components lend themselves to their most common final combustion products,  $\text{CO}_x$ ,  $\text{NO}_x$ , and  $\text{H}_2\text{O}$ , in gaseous form due to the high enthalpy of reaction.<sup>1,2,24</sup> These energetic materials oftentimes suffer from inefficiencies stemming from their stoichiometric imbalance since the stability of the compounds and molecular shape may limit complete combustion.<sup>6,21,22</sup> Furthermore, the gaseous nature of the CHNO energetic material’s final products guarantees a plateau in performance and energy release because the energy that went into gasifying these compounds is not recovered through condensation.<sup>1,6,22,25</sup> Typical adiabatic flame temperatures of these materials (not considering void collapses in the crystal structures) are  $\sim 2000$  K and have propagation velocities on the order of km/s.<sup>2,6</sup>

Alternatively, heterogeneous energetic materials are synthesized by combining independent fuels and oxidizers to achieve stoichiometric balances, thus improving overall efficiency from a chemical conversion standpoint.<sup>6,22</sup> The first energetic materials (e.g. gunpowder as discussed in section 1.1.1) were actually heterogeneous energetic materials. Heterogeneous energetic materials are commonly associated with propellants and pyrotechnics since they combine different materials as the fuels and oxidizers (e.g. aluminum, ammonium perchlorate, and HTPB in the solid rocket motors for the Space Shuttle).

For this dissertation, research will be focused around metal-based heterogeneous energetic materials which are synthesized by physically mixing fuels and oxidizers for controlled stoichiometry. Thermites are an example nonexplosive sub-class of heterogeneous energetic composites that are characterized by the rapid



reduction-oxidation reaction between a metal and oxidizer (which is also typically metal-based).<sup>6</sup> Metals have been found to be useful fuels in a variety of heterogeneous energetic material configurations, though, including in solid rocket propellants and pyrotechnics where the oxidizer could be air or even a fluoropolymer. Broadly speaking, metallized energetic materials have a higher gravimetric/volumetric energy density, heat of reaction, and adiabatic flame temperature (2000-4000+ K) compared to organic energetic materials since the final products are in the condensed phase and partially recover the heat of condensation (see Figure 1-2).<sup>1,5,6,22,24-31</sup> However, heterogeneous energetic materials have a significant disadvantage since they are oftentimes prepared by mixing relatively large particles (nm- $\mu$ m) and, since the characteristic diffusion time between fuel and oxidizers has increased, the reactions occur on a substantially longer timescale.<sup>6,24,27,32,33</sup> As a result of the coarser mixing between fuel and oxidizer, these materials have propagation velocities ranging from cm/s to m/s.<sup>34-36</sup>



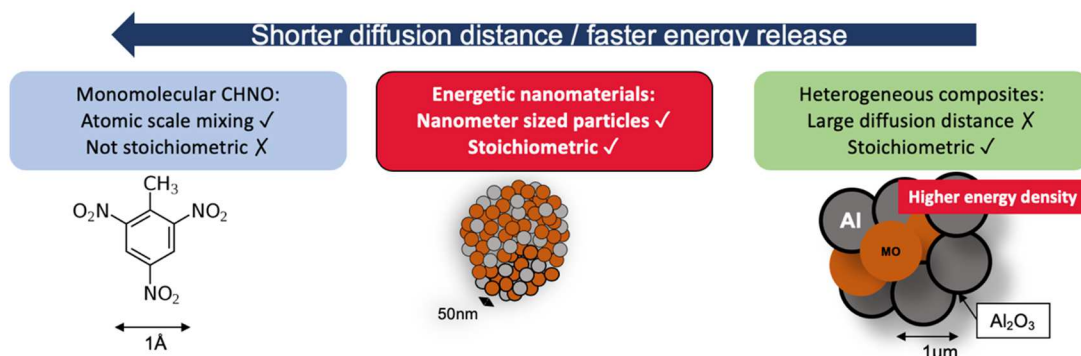
**Figure 1-2:** Enthalpy of reaction on a volumetric and gravimetric basis of common monomolecular explosives and thermite mixtures.

## ***1.2 Energetic nanomaterials***

The fundamental differences between homogeneous and heterogeneous energetic materials lie within their reactivity and, in order to make heterogeneous competitive candidates for energetics applications, it is necessary to begin investigating ways to improve their reactivity. As explained in the previous sections, the primary detractor from heterogeneous energetic combustion is the diffusion distance between the reactants – diffusion distances would need to be reduced by orders of magnitude to even approach propagation rates of homogeneous energetics. Nanomaterials, which have sub-micrometer characteristic lengths, are an attractive option for reducing the diffusion distance between the fuel and oxidizers in heterogeneous energetics.

Over the past few decades, the rise of nanomanufacturing has enabled research on improving reactivity of heterogeneous energetics to combine the desirable characteristics of traditional thermites (i.e. high heat of reaction) and monomolecular energetic compounds (i.e. short characteristic diffusion time). By mixing solid fuels and oxidizer powders with diameters on the nanometer scale, novel improvements in burn rate accompanied the newly synthesized “energetic nanomaterials” or “metastable intermolecular/interstitial composites.”<sup>6,22,24,27,32,33,37</sup> Physically mixed nanothermites, which are often prepared by intimately mixing metal/metal oxide powders, demonstrated an improvement in reactivity over the traditionally mixed micron-sized powder thermite (see Figure 1-3).<sup>6,38</sup> The improved reactivity of heterogeneous energetics by incorporation of nanomaterials have since led to widespread investigation and incorporation of the materials into novel nanothermites and metallized propellants.<sup>24,25</sup> However, while the apparent reactivity saw dramatic improvements,

new limitations that were not significant in thermites with micron-scale particles became hindrances to the initiation of the reaction and its ultimate conversion, as will be discussed in later sections.<sup>24,39,40</sup>



**Figure 1-3:** Schematic showing the relative size, scales, and characteristics of homogeneous energetics, heterogeneous energetics, and metastable intermolecular composites for comparison purposes.

Research areas on energetic nanomaterials can be divided primarily between chemical or physical modifications to either improve reactivity of composites or uncover structure-function relationships in the materials for improved tailoring of energy release.

### 1.2.1 Mechanisms of energetic nanomaterial combustion

The reaction mechanism of nanoenergetic composites has been studied extensively, albeit with some heavily debated conclusions. The mechanisms by which these composites react is a convoluted problem concerning heat transfer, mass transfer, and reaction kinetics.<sup>1,6,11,24,32,37,38,41–47</sup>

#### 1.2.1.1 Metal particle combustion

A variety of metals have been used as part of nanoenergetic materials mixtures including boron<sup>48</sup>, magnesium<sup>49,50</sup>, silicon<sup>51,52</sup>, and titanium<sup>53–55</sup>; however aluminum nanoparticles have been the most widely employed fuel in high performance

formulations due to their availability, low cost, and relatively high energy density as discussed in the literature.<sup>56</sup>

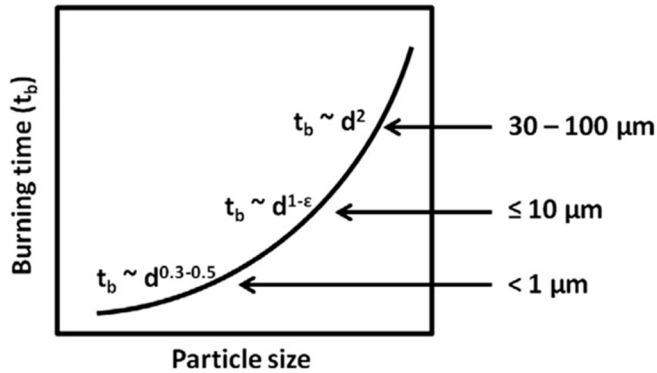
Looking back at the discussion on reaction kinetics and transport phenomena (see section 1.1.2), the reaction rate ( $\dot{\omega}$ ) of a component is a critical parameter in energetic material combustion since it is directly related to the propagation rate (and therefore mass burning rate) of a composite. For particle systems, the reaction rate at the surface of the particle can be generally expressed by Equation 1-15 where A is the pre-exponential factor for the reaction rate,  $\beta$  is the oxidizer mass flux, k is the temperature-dependent reaction rate, and  $C_i$  is the concentration of species “i.”<sup>57</sup> The oxidizer mass flux can be calculated by Equation 1-16 where Sh is the Sherwood number, D is the mass diffusivity, and d is the particle diameter.<sup>57</sup>

$$\dot{\omega} = A \left( \frac{\beta k}{\beta + k} \right) C_i \quad (1 - 15)$$

$$\beta = Sh \left( \frac{D}{d} \right) \quad (1 - 16)$$

The combustion of the particles can therefore be divided into two regimes: a diffusion-limited regime ( $k \gg \beta$ ) and a kinetic-limited regime ( $k \ll \beta$ ).<sup>57</sup> In the diffusion limited case, the burn time of the particle scales with the square of the particle diameter ( $\tau_{diff} \propto d^2$ ) whereas the burn time scales directly with the particle diameter in the kinetically limited case ( $\tau_{kin} \propto d$ ).<sup>57</sup> Studies that wished to evaluate the burn time of aluminum particles as a function of particle diameter found that large particles (30-100  $\mu\text{m}$ ) were burning in the diffusion limited regime and particles  $\leq 10 \mu\text{m}$  in diameter burned in the kinetically limited regime (see Figure 1-4).<sup>58</sup> This result is in good agreement with the premise that reducing particle size increases reactivity by

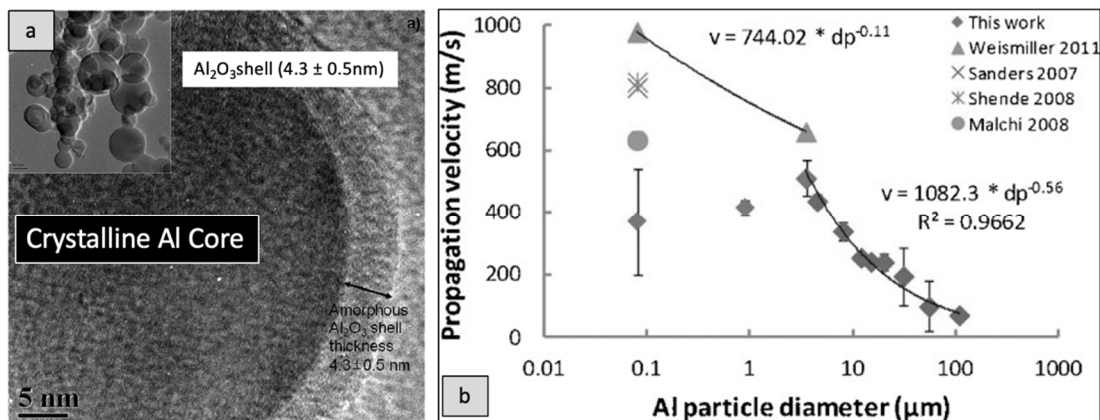
reducing diffusion distances and therefore the only barrier to faster combustion is the kinetics of the reaction itself. However, for very small particles  $<1\ \mu\text{m}$  in diameter, the relationship is actually sub-unity (i.e.  $t_{burn} \propto d^{0.3-0.5}$ ) which could not be explained well by current theory and is still a point of concern for using nanomaterials.<sup>38,58</sup>



**Figure 1-4:** Data summary for burn time of metal nanoparticles and its dependency on particle size. Adapted from Chakraborty et al.<sup>58</sup>

A possible explanation for the unusual particle burning behavior may lie in the morphology of the particles at the nanoscale. Metal nanoparticles like those used in energetic materials suffer from a significant drawback: when examined under an electron microscope and in thermogravimetric analysis, one notices that the nanoparticles are coated with a 2-5 nm layer of oxidized metal (e.g.  $\alpha\text{-Al}_2\text{O}_3$ ,  $\text{TiO}_2/\text{TiN}$ ,  $\text{B}_2\text{O}_3$ ) as shown in Figure 1-5a.<sup>24,42</sup> This oxidized shell, whether formed naturally or intentionally by the manufacturer to preserve the core,<sup>24</sup> serves as a protective measure against the pyrophoric nature of the crystalline interior. Regardless of its origin, the passivating oxide layer ultimately reduces the active metal content available for the reaction in the mixtures and, these reductions become significant hindrances on performance (see Figure 1-5).<sup>40</sup> A study by Sullivan et al. partially attributed the underwhelming performance to the reduced active metal content, the relatively larger diffusion barrier for active metal content to overcome, and the heat sink introduced by

the passivating layer, thus the burn rates of nanoparticles do not increase as expected for such a considerable change in particle size<sup>38</sup>



**Figure 1-5:** (a) TEM image of as-received nanoaluminum fuel particles demonstrating the naturally passivating oxide shell.<sup>59</sup> (b) Plot of burning rate vs. fuel particle size. Adapted from Sullivan et al.<sup>38</sup>

There is a large scattering of results for studies looking to improve performance by using small particles. While some reports indicate that using small, nm-sized particles can improve performance in materials significantly,<sup>12</sup> others have found that these materials actually hinder performance with a negative exponent for diameter dependence for burn time.<sup>38</sup> With such widespread results, reaction mechanisms that are used to explain them are similarly scattered. To better understand the mechanisms of energetic nanomaterial combustion, the different types of reaction mechanisms are discussed below.

#### 1.2.1.2 Heterogeneous phase reactions

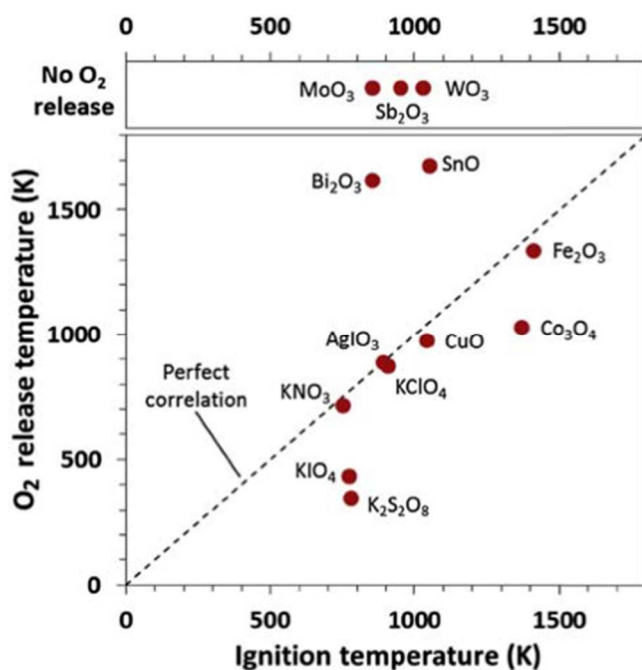
The most common oxidizers used in nanothermite formulation have traditionally been iron oxide (Fe<sub>2</sub>O<sub>3</sub>)<sup>60–62</sup>, copper oxide (CuO)<sup>63–66</sup>, bismuth oxide (Bi<sub>2</sub>O<sub>3</sub>)<sup>65,67–69</sup>, molybdenum oxide (MoO<sub>3</sub>)<sup>33,65,67,70–72</sup>, and tungsten oxide (WO<sub>3</sub>)<sup>61,73</sup>, however, more exotic oxidizers like potassium perchlorate (KClO<sub>4</sub>)<sup>55</sup>, iodine pentoxide (I<sub>2</sub>O<sub>5</sub>)<sup>54,74,75</sup>, fluoropolymers (PVDF, Viton, PTFE)<sup>34,76–78</sup> have also been incorporated

for more specialized purposes. In principle, any oxidizer could be used provided that the metal in the compound has a higher reduction potential than that of the metal fuel being used.<sup>24</sup>

An early theory of energetic nanomaterial combustion suggests that the reaction is controlled by the diffusion of gaseous byproducts from oxidizer decomposition during reaction. For example, during the decomposition of CuO or Fe<sub>2</sub>O<sub>3</sub>, one would see the release of gaseous oxygen as the oxidizer undergoes transformations into its suboxides (e.g. Cu<sub>2</sub>O, Fe<sub>3</sub>O<sub>4</sub>, etc.) and the release of gaseous oxygen is evidenced by a large plume (depicted in Figure 1-7a). The reaction would be initiated by the decomposition of the oxidizer and heating of the metal fuel simultaneously and the reactants would then diffuse towards each other.

To evaluate the validity of this mechanism, it would be reasonable to expect that the oxygen release temperature of common oxidizers would be below the ignition temperature of a given composite (with invariant fuel and equivalence ratio). Early experiments by Zhou et al. studied the role of oxygen release across the Al/CuO, Al/Fe<sub>2</sub>O<sub>3</sub>, and Al/ZnO systems through time-resolved, time of flight mass spectrometry (TOFMS) with a temperature-jump (T-Jump) system that achieved heating rates of  $\sim 10^5$  K/s found that those oxidizers that more readily released oxygen during decomposition exhibited greater levels of reactivity, thus supporting this hypothesis.<sup>79</sup> This conclusion was also in agreement with the heightened reactivity of Al/CuO in comparison to its counterparts when tested in constant volume combustion experiments.<sup>6,24,25,61,65,79</sup> However, later experiments by Jian et al. measured the oxygen release temperature of common oxidizers for aluminized thermites using the same T-

Jump/TOFMS system and found that, while some oxidizers ( $\text{CuO}$ ,  $\text{Fe}_2\text{O}_3$ ,  $\text{KClO}_4$ ) did release gaseous oxygen prior to the ignition temperature of the corresponding thermite mixture, many others actually did not ( $\text{Bi}_2\text{O}_3$ ,  $\text{WO}_3$ ) (see Figure 1-6).<sup>6,41</sup> In fact, some oxidizers did not release any gaseous oxygen despite there being a clear reaction with Al when incorporated into an nanothermite.<sup>41</sup> These experiments definitively demonstrated that gaseous oxygen was not a prerequisite for combustion in these materials and critically weakened the argument for the heterogeneous phase reaction mechanism.



**Figure 1-6:** Plot of O<sub>2</sub> release temperature in T-Jump/Time of Flight Mass Spectrometer vs. ignition temperature of the oxidizer when mixed stoichiometrically with aluminum nanoparticles. Adapted from Zarko et al.<sup>6</sup>

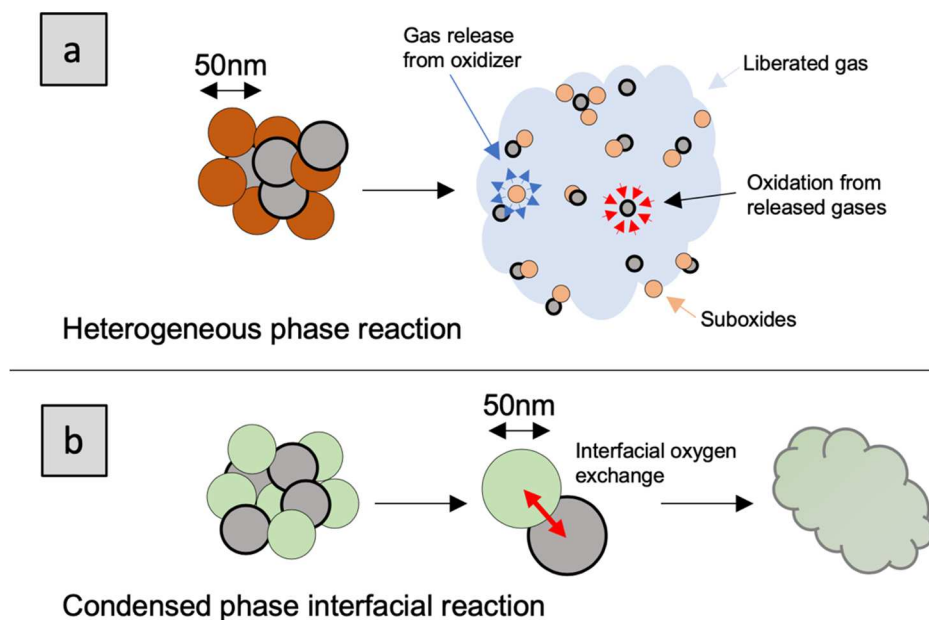
Despite the apparently simplicity of this reaction mechanism, it does not hold up universally across the studied energetic nanomaterials, thus suggesting there must be other mechanisms at play during combustion. This reaction mechanism may be of



importance in some materials and may play a role in the combustion behavior, though, and this concept is explored in Chapter 2.

### 1.2.1.3 Condensed phase interfacial reactions

Another combustion mechanism that is proposed for energetic nanomaterials considers condensed-phase transport of oxygen at the interfaces of the reactants (see Figure 1-7b). Conceptually, this is a much more difficult problem to solve since it requires knowledge of the interfaces and properties of the oxidizing shells. This mechanism does make sense for nanomaterials though, especially considering that nanoparticles have a high specific surface area and that composites made with them would have a high degree of interfacial contact.



**Figure 1-7:** Depiction of (a) heterogeneous phase reactions vs. (b) condensed-phase interfacial reactions.

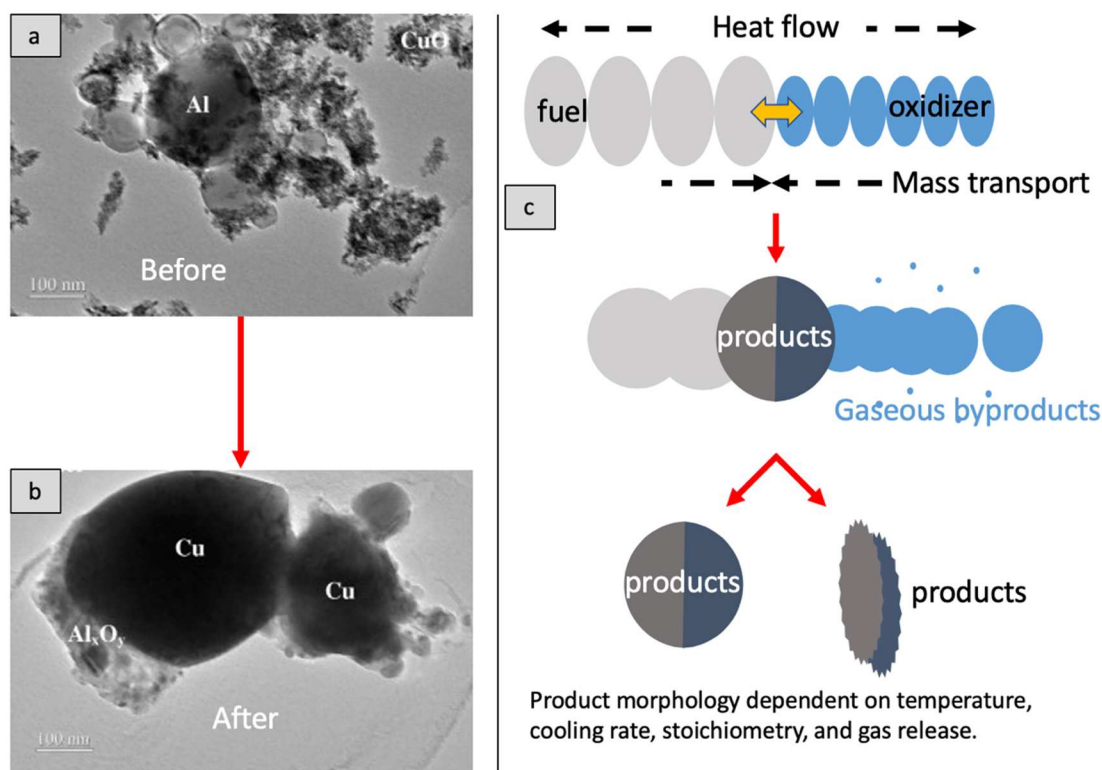
This reaction mechanism essentially suggests that the transport of oxygen in the condensed phase should be related to the ignition temperature of the composite. To study this, Wang et al. studied oxidizers with varying crystal structure, and thus

different oxygen transport properties, and found that ignition temperature could be related to the characteristics of the oxidizer.<sup>69,80</sup> In their experiments where  $\text{Bi}_2\text{O}_3$  was doped with Al, C, and Ta, when the oxygen vacancy increased and the metal-oxide bond energy decreased, the oxygen ion conductivity increased while the ignition temperature decreased.<sup>69</sup> Follow-up studies by the same author also demonstrated the relationship between average bond energy and electronegativity with ignition temperature in Al-based doped perovskite mixtures.<sup>80</sup> These results gave significant credibility to this reaction mechanism, but the method has not yet been applied to some oxidizers in Figure 1-6.

A more serious problem posed by this mechanism concerns the argument for using nanomaterials – this reaction mechanism implies there must be a loss of nanostructure at the interfaces of energetic nanomaterial composites. The exothermicity of combustion reactions would result in large temperature rises in the composite and, since nanoparticles are known to agglomerate easily and can sinter together, it is reasonable to ask which happens first: the reaction or the loss of nanostructure?

The loss of nanostructure in energetic composites resulting from high temperatures and interfacial reactions is called reactive sintering.<sup>47</sup> As depicted in Figure 1-8a, the process has been observed at the nanoscale using a high heating rate stage during transmission electron microscopy (TEM).<sup>47</sup> Simulations of sintering in these composites reveal that this process can initiate in as little as ~50 ns – much faster than the burn times of the composites.<sup>58</sup> Reactive sintering negatively impacts the advantages of using nanomaterials since it erases the nanostructure that would

otherwise lend itself to a high degree of interfacial mixing between reactants. Unfortunately, reactive sintering can prematurely arrest reactions in energetic nanomaterials since the reactants may become segregated as the reaction proceeds. This process may account for the unusual size dependency for burn time of small particles, too. Ultimately, reactive sintering is a major cause for incomplete reactions and the diminished returns in reactivity when using nanomaterial components in energetic materials.



**Figure 1-8:** Images demonstrating reactive sintering which can negatively impact reactivity in energetic materials. (a) TEM image of a physically mixed nano Al/CuO energetic composite. (b) Image of the nano Al/CuO energetic composite after heating on a fast heating stage TEM. (c) Depiction of reactive sintering process. Adapted from Sullivan et al.<sup>47</sup>

The reality is that energetic nanomaterials combustion is likely an amalgam of the heterogeneous phase reactions and condensed phase reactions as soon as the

reaction is started. While interfacial reactions may lend an explanation to the ignition characteristics of the composites, heterogeneous reactions must be occurring for oxidizers that release gaseous oxygen. The interplay of these different types of oxidation mechanisms has been explored before by Sullivan et al.<sup>61</sup> and will be directly observed in this dissertation.

#### *1.2.1.4 Melt dispersion mechanism*

Another possible mechanism proposed by Levitas et al. is the melt dispersion mechanism, which assumes a reaction limited by mass transfer and proposes that pressure built up by an expanding Al core leads to a catastrophic and explosive failure of the oxide shell.<sup>45</sup> This failure subsequently allows for the reaction between the fuel and the oxidizer.<sup>45</sup> Although this mechanism holds up in simulations and seemingly explains the “burst” shells of  $\alpha$ -Al<sub>2</sub>O<sub>3</sub> in microscopy images of nanothermite products, the heating rate required to achieve such an explosion would exceed even the most sophisticated heating schemes (other than a laser).<sup>11,24,45,81</sup> Alternatively, since Al is soluble within the  $\alpha$ -Al<sub>2</sub>O<sub>3</sub> shell and the melting temperature of the crystalline phase is far below that of the shell, it could be reasonably assumed that molten Al diffuses through the shell where it is left to react with the oxidizer on the surface of the nanoparticle.<sup>81</sup> Upon reacting with the oxidizer, the mixture could reach the adiabatic flame temperature (2000-4000 K)<sup>26,31</sup> locally, thus exceeding the melting temperature of the  $\alpha$ -Al<sub>2</sub>O<sub>3</sub> (2326 K), leaving it to disintegrate and expose the remainder of the core materials. In either scenario presented, the passivating oxide layer on the exterior of the metal nanoparticle core is the rate-limiting step in nanothermite reactions.

### 1.2.2 New methods to tailor energetic nanomaterial combustion

#### 1.2.2.1 Variations to nanoparticle structure

Seeing as the structure of the metal fuel particle can dramatically impact performance of energetics, multiple studies have been performed to safely preserve the active content without compromising reactivity. In some attempts to reduce the impact of the oxide shell, the aluminum nanoparticles were encapsulated in polymer matrices<sup>82</sup> or protected with polytetrafluoroethylene (PTFE) after nanoparticle generation;<sup>83</sup> however, these cases oftentimes resulted in reduction of the active Al content by mass below that of the starting material. Other studies have attempted to prepare energetic nanomaterials by replacing the  $\text{Al}_2\text{O}_3$  shell on the metal core with the intended oxidizing compound; however, this technique has resulted in minimal improvements because surface reactions that produce a partially oxidized shell still occur.<sup>84</sup>

While many of the attempts to reduce the impact of a passivating layer on the reaction dynamics have found preliminary success in formulation, the lack of energetic material content per unit mass and the dependency on starting materials act as caveats to their success and make them unlikely to pioneer the next push in energetic nanomaterial research. Furthermore, since studies have shown evidence of a transport-limited reaction within energetic nanomaterials that is primarily hindered by the presence of a passivating oxide shell on the fuel, it is desirable to have careful control over the morphology of stabilized fuel nanoparticles. Metal nanoparticles have previously been made through either physical milling of larger primary particles or through aerosol synthesis routes, both of which are time and energy intensive and have the tendency to produce inconsistent products in regards to size distribution and active

metal content.<sup>24</sup> However, recent advancements in nanoparticle synthesis have been realized through the rapid heating of a metal-laden material that can sustain temperatures above the metal's melting point without decomposing. In a recent publication by Yao et al., metal nanoparticles (sub-5nm) with a narrow size distribution were successfully synthesized from micron-sized primary metal nanoparticles that were encapsulated within a rGO matrix.<sup>85</sup> These resultant pure crystalline Al nanoparticles are considered stable due to the high thermochemical stability of the surrounding rGO material.<sup>85</sup> While nanoparticle synthesis via this route is superior to those previously discussed, the mass ratio of rGO to active Al content within the sheets has only been studied up to loadings of 20:1 rGO:Al, which is still below the usable threshold in an energetic composite. Chapter 3 in this dissertation will explore this method further with increased metal content with possible application for energetic nanomaterials.

#### *1.2.2.2 Investigating the role of architecture*

Similar to the way that energetic nanomaterials were brought into the spotlight by exploiting different material architectures to improve performance, advanced manufacturing methods are opening up new routes to increase heat transfer or modulate chemistry to enhance versatility.

Additive manufacturing has come to the forefront of these architecture modulation studies with the development of electrospraying, electrospinning, and 3D printing. A common characteristic of these studies is the incorporation of energetic materials into a fluoropolymer-based reactive ink that is then deposited onto a surface where the ink solvent is then evaporated away. Researchers have studied the role on

the type of additive manufacturing method employed<sup>86,87</sup>, explored a wide variety of polymers<sup>35,77</sup> and polymer configurations (e.g. ink-based vs. solid melting/cooling<sup>76</sup> and heated bed vs. light curing<sup>88</sup>), and have even developed advanced printing methods that can introduce chemistry variations on-the-fly.<sup>89,90</sup> Others have used synthesis methods like electrospraying to produce reactive particles that have the fuels and oxidizers in an enhanced interfacial contact.<sup>83,91</sup> Many of the reaction mechanisms detailed in these studies focus on the role that the porosity and energetic content of the reaction plays on heat transfer and chemistry as it is related to propagation. Since this is a relatively new area of research for energetic materials, it is important to improve understanding of reactivity and propagation in these types of composites.

#### *1.2.2.3 Incorporation of additives*

The simpler approach to modulating energetic material reactivity that has been widely pursued is introduction of additives that can modify reaction rates and heat transfer rates. To be considered an “additive” in the composite, the newly introduced material is typically only present as a minor component in the overall mass/volume of the composite. There is a seemingly infinite number of additives that have been explored for energetic material modulation, but the additives are generally speaking either chemically active or inert. Chemically active additives are oftentimes introduced as either a catalyst<sup>92</sup> or a complementary fuel/oxidizer<sup>54,93</sup> which increases the energy release rate of the composite or improves the efficiency of the reaction. These studies typically contain detailed reaction kinetic analysis with respect to the new additive’s role on the chemistry, though this type of analysis is often difficult given the ultrafast nature of the reaction. Meanwhile, the studies on inert additives that have been

incorporated into energetic materials to the studies typically fall back on changes to heat transfer, such as with the incorporation of materials with high or low thermal conductivity.<sup>94–97</sup> While not new, this avenue of energetic material modulation has also been of interest lately since new manufacturing methods offer novel ways prepare additives and control their placement in the composite structure.

### ***1.3 Motivation and research outline***

Energetic nanomaterials are promising candidates for enhanced energy release in highly tailorable, easily manufacturable composites. Research in energetic nanomaterials has begun turning towards incorporation of these materials into novel solid rocket propellants, especially since solid propellant technology has remained rather stagnant over the past few decades and any increase in energy release rates can improve motor performance and cut costs. For context, it costs ~\$10,000/pound of payload to reach earth orbit, so advancement in propulsion technology is paramount given the recent revival in interest space exploration. Additive manufacturing techniques like electrospray, electrospinning, and 3D-printing also lend themselves to advanced architecturing of solid propellants which could incorporate energetic nanomaterials to make a new, customizable class of solid rocket propellants. With constant improvements to the tunability and performance of these materials through careful fuel, oxidizer, and structural manipulations, specialized devices can be realized as part of a major development in the rather stagnant world of solid rocket propellants and motors.

This dissertation attempts to uncover and quantify structure-function relationships in energetic nanomaterials by modifying chemical and physical properties



of the materials and characterizing the observed changes using new diagnostic tools. The new technologies will then be applied to the development of new energetic nanomaterials with tailorable performance and enhanced versatility.

#### 1.3.1 Developing diagnostic tools for characterizing energetic material combustion

Development of new diagnostic tools is critical to advancing the understanding of structure-function relationships in energetic nanomaterials. Seeing as temperature and propagation rate are of particular interest for energetic material combustion, new diagnostics must be prepared to analyze both. New diagnostic tools developed in this dissertation can be used to estimate combustion temperatures with  $\mu\text{s}$  resolution using optical methods. In particular, a high-speed color camera pyrometer has been developed to capture the spatiotemporal dynamics of energetic material combustion. The tool is then later modified into a high-speed microscope that enables measurements at high spatial ( $\sim\mu\text{m}$ ) and temporal ( $\sim\mu\text{s}$ ) resolutions and can be used to observe reaction dynamics at the near-particle scale. The microscopy apparatus and accompanying image analysis software also allow for measurements on reaction front structure evolution in solid propellants to probe how structural changes in a propellant may impact propagation. Further developments with this tool also enabled measurement of thermophysical properties and activation energies of reactions using in-operando temperature measurement methods.

#### 1.3.2 Probing the role of chemical formulation energetic material combustion

Having developed a method by which to directly observe reaction dynamics of energetic nanomaterials, attention then turns towards probing the role of chemical

formulation on reactivity in energetic materials. The first study in this dissertation probes the role of gas generation vs. thermal effects in energy release rate in an attempt to improve combustion efficiency by mitigating reactive sintering with the introduction of a gas-generating oxidizer that prevents materials from coalescing. After exploring the role of different oxidizers to improve tailorability and performance in energetic nanomaterials, attention will be turned towards exploring new, scalable methods to manufacture fuels with increased active metal content. This research effort was ultimately unsuccessful in improving energetic content though and the poor performance was attributed to reactive sintering. Seeing as reactive sintering is a significant hinderance in performance of energetic nanomaterials, the high-speed microscopy apparatus was used to quantify the process in-operando and speculate methods to improve efficiency. Methods to overcome inefficiencies and effectiveness of those methods is further explored in later experiments by drawing upon the energy transfer concepts introduced in section 1.1.2.

### 1.3.3 Probing the role of physical architecture on energetic material combustion

The role of physical architecture on propellant combustion is also investigated with the aim of improving efficiency and versatility in solid propellants. Propagation rate (and therefore thrust) is directly tied to heat transfer in the composite and energy liberated during the combustion, therefore it is important to improve understanding of how different modes of heat transfer can be utilized to modulate performance. One study will investigate the role of solid propellant additives with contrasting heat transfer properties on propagation rate. This dissertation will conclude with the development of a novel solid rocket propellant that can be remotely ignited using microwaves, setting

the stage for a remotely staged solid propellant architecture that would provide unprecedented control over solid propellant combustion.

## Chapter 2: High-speed 2-dimensional temperature

measurements of nanothermite composites: Probing thermal vs. gas generation effects

### *2.1 Summary*

This work investigates the reaction dynamics of metastable intermolecular composites through high speed spectrometry, pressure measurements, and high-speed color camera pyrometry. Eight mixtures including Al/CuO and Al/Fe<sub>2</sub>O<sub>3</sub>/xWO<sub>3</sub> ( $x$  being the oxidizer mol. %) were reacted in a constant volume pressure cell as means of tuning gas release and adiabatic temperature. A direct correlation between gas release, peak pressure and pressurization rate was observed, but it did not correlate with temperature. When WO<sub>3</sub> was varied as part of the stoichiometric oxidizer content, it was found that Al/Fe<sub>2</sub>O<sub>3</sub>/70% WO<sub>3</sub> achieved the highest pressures and shortest burn time despite a fairly constant temperature between mixtures, suggesting an interplay between the endothermic Fe<sub>2</sub>O<sub>3</sub> decomposition and the higher adiabatic flame temperature sustained by the Al/WO<sub>3</sub> reaction in the composite. It is proposed that lower ignition temperature of Al/WO<sub>3</sub> leads to the initiation of the composite and its higher flame temperature enhances the gasification of Fe<sub>2</sub>O<sub>3</sub>, thus improving advection and propagation as part of a feedback loop that drives the reaction. Direct evidence of such gas release promoting reactivity was obtained through high speed pyrometry videos of the reaction. These results set the stage for nanoenergetic materials that can be tuned for specific applications through carefully chosen oxidizer mixtures.

## 2.2 Introduction

Research in nanoscience, as in many fields, has permeated the development of energetic materials where the demand for improved reactivity may be achieved with increased intimacy between reactants. Traditional monomolecular CHNO systems represent optimal reactant proximity, with mixing achieved at the molecular scale. However, the gaseous nature of their reaction products allow limited enhancements in their energy content.<sup>21,22</sup> As a result, much recent research has been directed towards nanoscale composite energetic materials incorporating metal nanoparticles as the fuel so as to enhance energy content and release rate.<sup>27</sup> One of the primary advantages of using metallic fuels is their high volumetric energy density leading to some metal based energetic materials having heats of reaction a few times larger than the state of the art CHNO systems such as High Melting Explosive (HMX), hexanitrohexaazaisowurtzitane (CL-20), etc.<sup>21</sup> However, using non-molecular, fuel-oxidizer systems implies a diffusion-limited process. It has been demonstrated repeatedly that the use of nanoparticles, with its decreased length scale, leads to a metastable system with orders of magnitude improvement in reactivity compared to their counterparts mixed at coarser scales.<sup>98</sup> A large body of recent research has been directed at exploring the underlying mechanism responsible for the fast reaction rates observed in such Metastable Intermolecular Composites (MIC)/nanothermite combustion, with the general consensus being that the dominant energy transfer mechanism in MIC powder is convection and molten particle advection<sup>11</sup> and the primary mode of ignition/reaction is via the condensed phase diffusion of reactant moieties.<sup>47,99</sup>

A significant benefit of MIC systems is their tunability, which stems from the extensive permutations in the selections of fuels, oxidizers,<sup>41,100</sup> gas generators<sup>101</sup> and architecture<sup>102,103</sup>, with the state of the art systems showing flame speeds as high as 4000 m/s.<sup>104</sup> Several methods have been developed to quantify their reaction dynamics with, ignition temperature and speciation measurements,<sup>41,60</sup> thermo-analytical methods,<sup>105</sup> flame speeds,<sup>37</sup> thermo-equilibrium software (CEA, Cheetah)<sup>31</sup> and constant volume combustion<sup>65</sup> being the most common. Although these methods do help in quantifying the combustion characteristics of MIC's, the flame speed and pressure measurements are significantly affected by variations in their experimental design. Temperature, on the other hand, is a fundamental thermodynamic property and is directly related to energy release, although its measurement in energetic materials research is not as prevalent as one might suppose. Primarily, the wide range of reaction times (10 $\mu$ s - 100ms), temperature range (800K – over 4000K) and spatial inhomogeneity make robust temperature measurements a challenge. Moreover, MIC's have significant emission interference from atomic and molecular species that participate in the high temperature reaction, thereby reducing the applicability of broadband measurement techniques.

Despite these limitations and challenges, the non-invasive aspect and relative simplicity of emission measurements as compared to other optical methods and the artifacts introduced by traditional invasive methods mean that optical emission is preferred for MIC characterization. Weismiller et al.<sup>62</sup> studied three different nanothermite compositions in an unconfined pile, and confined burn tube using multi-wavelength pyrometry. Kappagantula et al.<sup>106</sup> also examined several Al/CuO and

Al/PTFE based composites using an IR camera with metal additives as a means to optically measure performance. Despite the assortment of temperature measurement methods used to observe these energetic compounds, a common theme in each experiment was the inability for the compound to achieve its adiabatic flame temperature. The similarity across different composite mixtures has supported the conclusion that the systems under study were limited by the melting and decomposition of the oxide. Through such insight into the reaction dynamics, it might be possible to tune the reactivity of composite materials by altering the participation of competing reactions through mixture content variation. Prior work by Sullivan et. al<sup>61</sup> first explored this possibility by incrementally adding nanoscale  $\text{WO}_3$  to an  $\text{Al/Fe}_2\text{O}_3$  nanothermite to demonstrate an increase in pressurization rate under constant volume combustion environment. The result was counterintuitive as gas generation was observed to increase with the reduction of gas generating species ( $\text{Fe}_2\text{O}_3$ ) in the composite. The  $\text{Al/Fe}_2\text{O}_3$  nanothermite was postulated to be rate limited by oxidizer decomposition, and the performance improvement was attributed to efficient decomposition of  $\text{Fe}_2\text{O}_3$  induced by the heightened flame temperature at higher  $\text{WO}_3$  concentrations. While the results were corroborated with equilibrium calculations, no direct temperature measurement was made to support the claims regarding the interplay of reaction mechanisms. The purpose of this work is to extend the results of Sullivan et. al<sup>61</sup> through high-speed, multi-wavelength pyrometry to probe the tunable reactivity of a nanothermite composite in a closed pressure vessel. Moreover, the effects of enhanced gas production on reactivity is visualized under unconfined conditions with high spatial and temporal resolution using high speed color camera pyrometry.

## 2.3 Experimental

### 2.3.1 Materials and sample preparation

Commercially available aluminum nanoparticles (ALEX, Argonide Corp.) with an active content of 68.7 wt. % (from thermogravimetric analysis) and an average particle size of 50 nm were used in this study. The oxide nanopowders (CuO, WO<sub>3</sub> and Fe<sub>2</sub>O<sub>3</sub>) were procured from Sigma Aldrich and all had average diameters < 100 nm. The samples were prepared by dispersing a known amount of oxide in 10 mL of hexane and sonicated in an ultrasonic bath for an hour. This was done in order to break down the soft aggregates for ensuring better mixing with the fuel. A stoichiometric amount of aluminum was then added to this slurry and further sonicated for an hour. The slurry was left overnight to dry. The dry sample was gently scraped off the vial and broken up using a grounded spatula until powder consistency was achieved. Samples prepared included stoichiometric blends of Al/CuO, Al/Fe<sub>2</sub>O<sub>3</sub> and Al/WO<sub>3</sub>. Additionally, a set of 5 samples were made where we systematically adjusted the oxidizer composition by adding 20, 60, 70, 80 and 90% by mole of WO<sub>3</sub> to Fe<sub>2</sub>O<sub>3</sub> system, while maintaining the overall stoichiometry of the blend as outlined in the prior work by Sullivan et al.<sup>61</sup> For experiments involving high-speed videography, the slurry was kept as is for drop casting on a fine platinum wire for rapid heating, as discussed in the forthcoming sections.

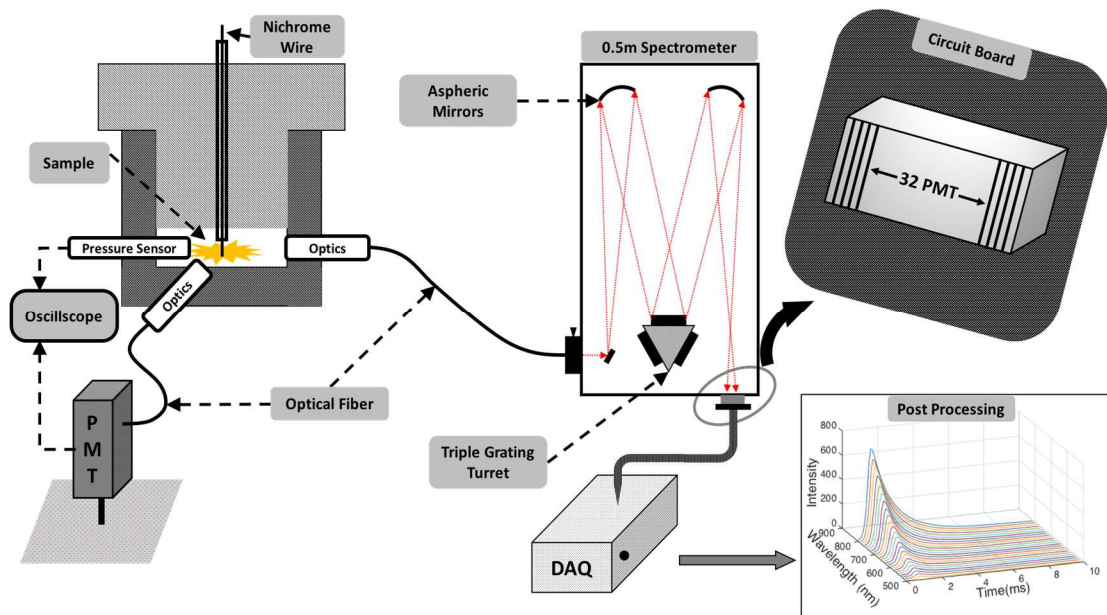
### 2.3.2 Constant volume combustion cell

The constant volume pressure cell used is described in detail elsewhere.<sup>61,107</sup> Briefly, the cell is a closed reaction vessel with a free volume of ~20 cm<sup>3</sup>, equipped



with 3 ports (as shown in Figure 2-1). One port houses a high frequency pressure transducer (PCB Piezoelectronics) for measuring the pressure signal generated during sample ignition and combustion. The second port is connected to an optical assembly used for collecting broadband emission from the inner edge of the vessel. The optical assembly consists of a plano convex lens (Thorlabs) which collects and focuses the light from the vessel into a 2 m long fiber optic cable (Dia. 1mm, Thorlabs Inc.) coupled to photomultiplier tube (PMT) (Hamamatsu). An optional neutral density filter (Thorlabs Inc.) is placed between the lens and the fiber for significantly brighter samples so as to prevent saturation of the detector. Both the PMT and the pressure transducer outputs are connected to a digital oscilloscope (Teledyne LeCroy Wavesurfer 3000) sampled at 5 MHz. The PMT's cathode voltage is selected empirically by monitoring the output current during trial tests. The third port houses another optical assembly which collects and relays the emission to a spectrometer, which is detailed in the next section. The cell was placed inside an artificial environment bag which was purged with argon to mitigate the influence of atmospheric oxygen on the reaction.

Each sample (25 mg) was tested in triplicate by ignition with a resistively heated nichrome wire connected to a DC power supply. A custom square wave generator was built in-house to simultaneously trigger the spectrometer, oscilloscope and the power supply for the nichrome wire. Owing to the high temperatures and significant production of gas and condensed species, a sapphire window was used to protect the optical assembly and was cleaned periodically so as to ensure accurate measurement of the emission.



**Figure 2-1:** Schematic of the experiment consisting of pressure cell and attached diagnostics. The pressure cell is shown on the left. The spectrometer is coupled to the pressure cell via a 1 m long optical fiber. The light from the fiber is spectrally dispersed by the selected grating on the turret which is subsequently imaged on the 32 channel PMT and digitized using the DAQ (Vertilon IQSP580). The digitized data is processed to produce time resolved spectra.

### 2.3.3 High speed 32-channel spectrometer

The optical assembly interfacing the spectrometer with the pressure cell consists of a Sapphire window, a Plano Convex lens  $1f$  from the inner edge of the vessel, a neutral density filter (ND2) and a 455 nm color glass filter for order sorting. The optical components are assembled inside a 0.5" lens tube and the end of the tube is closed with a SMA fiber optic adapter. The plano-convex lens collimates the light from the inner edge of the vessel which is subsequently attenuated by either an ND1 or ND2 filter depending on brightness of combustion. A 1mm diameter optical fiber (Princeton Instruments) transmits the emission into a 500 mm, triple grating, Czerny-Turner Imaging spectrometer (Acton), as shown in Figure 2-1. A 150 groove/mm grating and a slit width of 100  $\mu\text{m}$  was chosen for this work, which gave a dispersion

of 13 nm/mm at the focal plane and a spectral range of 464-867 nm. The primary objective of this work was to detect temperature, hence a low-resolution grating was chosen so as to obtain the widest spectrum.

The spectrometer outlet was coupled with a 32-channel PMT module (Hamamatsu, H7260) with each channel having a dimension of 0.8 mm x 7 mm (WxH), implying a spectral resolution of  $\sim 10$  nm/channel. A PMT-based detector was preferred in this work for its fast rise time (0.6 ns), tunable supply voltage (400-900 V), and single photon efficiency. The multichannel PMT (MC-PMT) is interfaced with a high-speed data acquisition (DAQ) system (Vertilon IQSP580) which terminates the MC-PMT to  $50\ \Omega$  and measures the current output at the anode with 14-bit resolution. The system is capable of sampling at a maximum rate of  $\sim 390$  kHz with an onboard storage for  $5 \times 10^5$  32 channel events. At  $\sim 400$  kHz, the current setup offers a good compromise between time resolution and total recording time, particularly for the combustion of nanothermites which are not only much slower than explosives, but also have a wide range of combustion time scales. One of the significant limitations of the setup is the  $20\ \mu\text{A}$  linearity limit of the MC-PMT, which limits the dynamic range of detection. This is offset by running multiple experiments where the ND2 filter, used for characterizing the emission at peak irradiances, was replaced with a ND1 for characterizing the cooler phases of the reaction.

Wavelength calibration for the 32-channel spectrometer was performed using an HgNe pencil lamp (Newport) and spectral response calibration was performed with a calibrated tungsten halogen lamp (Avantes HAL CAL) operating at 2440 K. To test the linearity of the MC-PMT, a high temperature blackbody source operated at 1273 K

was used and the signal was attenuated with neutral density filters (ND 0.4, ND 1). A detailed description of the calibration process and its results are provided in Section 2.8.1. In order to make quantitative temperature measurements, the corrected data from the MC-PMT was linearized<sup>108–110</sup> according to Wien's law as shown in Eq. 2-2, where  $\varepsilon$ : emissivity,  $h$ : Planck's constant,  $c$ : speed of light,  $K$ : Boltzmann constant,  $T$ : Temperature,  $L$ : Radiance,  $C_{1,2}$ : first and second radiation constant and  $\lambda$ : wavelength. Wein's Law is a derivative of Planck's Law (Eq. 2-1) and is applicable when the product  $\lambda T < 3000 \mu\text{mK}$ ,<sup>109</sup> satisfied by visible emission true for most energetic material combustion. Once the parameter  $Z$  is calculated, it is plotted against the wavelength and a straight-line fit is employed to calculate the temperature.

$$L(\lambda, T) = \frac{\varepsilon * 2\pi h c^2}{\lambda^5 \left\{ \exp\left(\frac{hc}{\lambda K T}\right) - 1 \right\}} \quad (2 - 1)$$

$$Z = \frac{\lambda}{C_2} \log\left(\frac{C_1}{\lambda^5 I}\right) = -\frac{\lambda}{C_2} \log \varepsilon + \frac{1}{T} \quad (2 - 2)$$

Implicit in the above linearization is the grey body assumption for the flame cloud generated by the reaction, of which there has been considerable debate in the community. Some authors have employed the  $\varepsilon \sim \lambda^{-1}$  dependence<sup>18,62</sup> which is derived from the spectral dependence of the absorption efficiency in the Rayleigh limit, while others have further corrected this correlation for the wavelength dependence in the absorptive index by using  $\varepsilon \sim \lambda^{-2}$ .<sup>111</sup> Recent efforts by Lynch et al. have suggested that under conditions where the high temperature flame front is optically thick, which presumably would be the case in a confined reaction, the gray body assumption could be valid, and hence is used in this study.<sup>112</sup> During the temperature fitting procedure, channels with spectral bandpass that overlapped prominent molecular emission, such a

Na doublet (588.95nm and 589.59nm) and AlO band ( $\Delta\nu = -1,0,+1$ , 464nm – 530 nm), were removed so as to improve the fit fidelity. Built-in MATLAB fitting routines were employed to estimate the final temperature along with error bounds thresholded at  $\pm 400\text{K}$ .

#### 2.3.4 Hot-wire ignition tests for spatiotemporal temperature maps

While the experimental setup including the spectrometer described above can measure the temperature of radiating particles using multi-wavelength techniques, its ability to measure the spatial dynamics of the combustion process is limited by means of its data acquisition. Owing to the nature of light collection through an optical fiber, the spectrometer would be biased to the brightest/hottest spots within the flame due to the exponential scaling of light intensity with temperature as per Stefan-Boltzmann Law. Furthermore, due to the spatially dynamic nature of the flame front, point light sources could constantly move in and out of the field of view, affecting the temperature measurement. As a complementary diagnostic to the spectrometer, a high-speed color camera is used to record videos of the combustion event so as to probe the highly dynamic flame front. A color camera operates by generating a mosaic of the scene under observation through selective bandpass transmission of light by the Bayer filter mask overlaid on the camera's CMOS sensor. This results in each pixel recording one color among red, green and blue. Further processing by the camera software 'demosaics' the data, generating RGB values at each pixel, to produce the image. Densmore et al. detailed how to employ a high speed color camera as a ratio pyrometer, which served as the basis for these experiments.<sup>15</sup>

To calibrate the Vision Research Phantom Miro M110, videos of both a blackbody source (Newport Oriel 6700 Series) and a calibrated lamp (Avantes) were recorded while varying temperature. MATLAB was used for pixel extraction and then demosaiced using the built-in functions. Ratios for theoretical pixel intensity were then calculated for the calibration temperature with Eq. 2-3 (where  $I_i$  is intensity of a channel,  $\chi_i$  is the spectral response of the channel as a function of wavelength,  $\psi_i$  is channel gain, and the Radiance  $L$  is given by Eq. 2-1). Calibration factors ( $C_{ij}$ ) were then calculated by comparing the measured values to the theoretical ones as shown by Eq. 2-4 (where  $C_{ij} = \psi_i/\psi_j$ ). A detailed description of the calibration process can be seen in section 2.8.2.

$$\frac{I_i}{I_j} = \frac{\psi_i \int L(\lambda, T) \chi_i(\lambda) d\lambda}{\psi_j \int L(\lambda, T) \chi_j(\lambda) d\lambda} \quad (2 - 3)$$

$$\left( \frac{I_i}{I_j} \right)_{exp} = C_{ij} \left( \frac{I_i}{I_j} \right)_{theor} \quad (2 - 4)$$

Raw pixel values were extracted from the video files using a custom MATLAB script and were passed to a processing pipeline which output the corrected intensities at each pixel. This sensor data was demosaiced using the built in MATLAB routine (gradient corrected demosaicing) to recover values for the red, green, and blue channels at each pixel, from which three ratios (green/red, blue/green, and blue/red), incorporating an inverse wavelength dependence of emissivity, were derived. To determine temperature, the calibration factors (from Eq. 2-4) that account for the spectral response of the sensor were applied to three ratios and matched to the theoretical curves with a minimization algorithm. Although color camera pyrometry does allow for spatiotemporal measurements of temperature, limitations in the method

by which raw data is collected are a strong source of error ultimately leading to deviations in temperature measurement from those reported by the spectrometer. Of the many elements that emit in the visible region during thermal relaxation, sodium is often the most noticeable with strong, persistent lines seen as a doublet at 588.95 nm and 589.59 nm.<sup>113</sup> Such emission contributes to the red and green channel intensities, due to their high spectral response at these wavelengths (Figure 2-8), leading to error in temperature calculations. Other elements that have strong emission and are possible sources of contaminants in the experiments performed include potassium and copper. Furthermore, the calculations detailed above fails to account for light scattering from small particulates that may be generated throughout the course of the reaction. Hence, in order to maintain fidelity, the error minimization algorithm used to calculate temperature is error thresholded to  $\sim 110$  K and the pixels that report higher errors are browned out in the final false-color images.

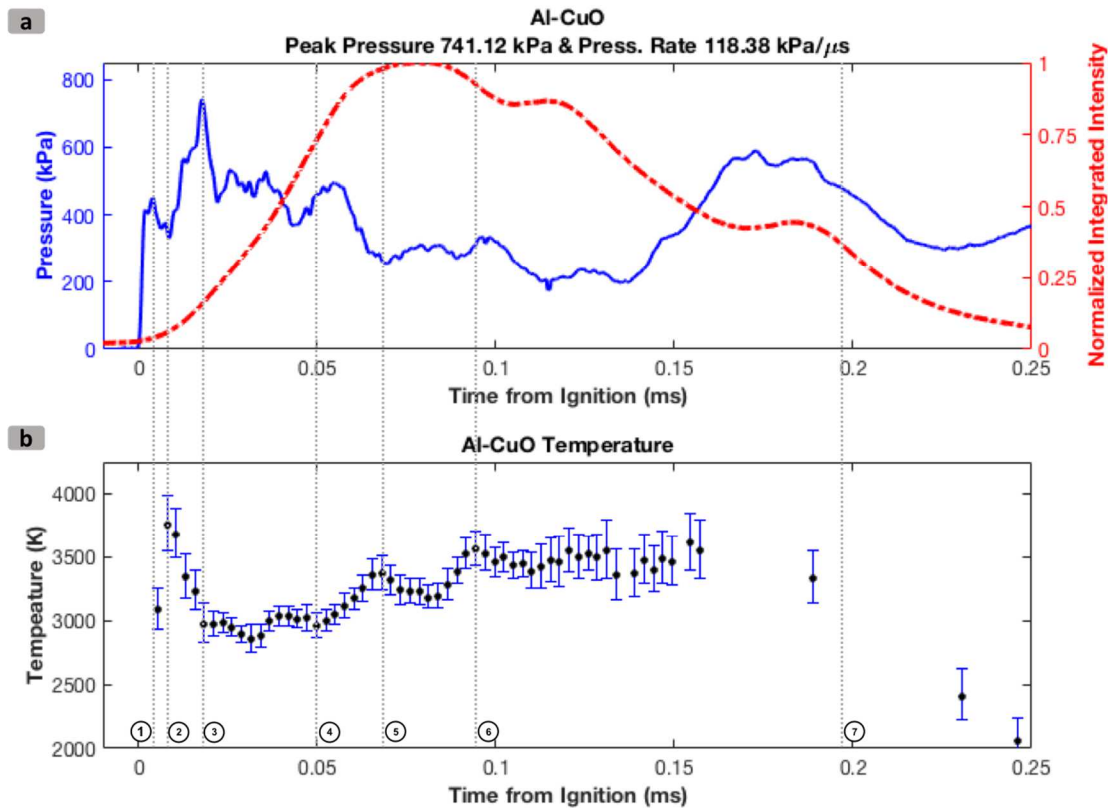
Wire ignition experiments were conducted in a stainless steel 6-way cross, with windows to visualize the combustion with the high-speed camera (Figure 2-10). The interior of the chamber was painted black in order to minimize light reflections. A premixed slurry of thermite sample were coated onto a 76  $\mu\text{m}$ -diameter platinum wire and resistively heated in a 1 atm argon environment for 4ms at a rate of  $\approx 10^5$  K/s using the T-Jump apparatus detailed by Zhou et al.<sup>60</sup> Two videos were recorded per sample at a framerate of 20,000 fps with the  $f/\#$  and exposure times chosen to provide the best signal to noise ratio.

## 2.4 Results

### 2.4.1 *Al/CuO nanothermite tests in pressure cell*

Al/CuO is the control system first examined here since it is the most studied thermite system. Figure 2-2 shows the temporal pressure response, normalized-integrated radiance across all PMT channels (Figure 2-2a) and the temperature fit (Figure 2-2b) for the Al/CuO reaction in the pressure cell. The peak pressure of the system is  $\sim 741$  kPa and the pressurization rate is  $\sim 118$  kPa/ $\mu$ s, calculated based on the rise time of the first pressure peak. The temporal temperature shown in Figure 2-2b is recorded using the ND2 Neutral density filter in order to quantify the emission at peak light intensity. The custom fitting algorithm enabled the simultaneous calculation of temperature, and the error associated with the fit which was thresholded to  $\pm 400$  K before plotting the profile shown in Figure 2-2b. The missing data points at longer durations correspond to such cases where the calculated error was higher than the threshold value. The figure is horizontally sectioned by gridlines so as to identify the different regimes to be discussed.





**Figure 2-2:** Temperature Measurement of Al/CuO nanothermite in pressure cell. a. Pressure-Normalized-Integrated Intensity, b. Temporal profile showing reaction temperatures. 1-2 Represents pressure drop; 2-3 Represents temperature drop and peak pressure; 3-4 Represents region of highest integrated intensity; 4-6 Represents region with sustained integrated intensity, increasing temperature, and declining pressure; 6-7 Represents temperature plateau and falling pressure.

Owing to the thresholding, the first temperature data point was obtained at  $\sim 0.005$  ms from ignition, where the integrated emission is approx. 15% of the peak integrated intensity. The pressure trace at this point corresponds to the first local maxima, as highlighted by the vertical line 1. Region 1-2 corresponds to a drop in pressure accompanied by a temperature rise to  $\sim 3600$  K which is followed by a sharp drop to  $\sim 2800$  K in region 2-3. This coincides with an increase in pressure to its maximum value at  $\sim 0.017$  ms (location 3) and the emission intensity is observed to have a sharp positive slope from point 2 onwards. At the instant of peak pressure, the

normalized emission is  $\sim 30\%$  and the temperature is near the adiabatic flame temperature for Al+CuO (2837 K).<sup>61</sup> Region 3-4 corresponds to the most substantial increase in integrated intensity as it rises from 30 to 80% at 0.05ms (location 4) although the temperature in this region is observed to plateau at  $\sim 3000$  K. The pressure profile, on the other hand, shows a steady decline in this region. Region 4-6 broadly corresponds to an increase in temperature to  $\sim 3500$  K (location 6) and the emission intensity is observed to first increase in the region 4-5 by  $\sim 20\%$ , attaining its peak at location 5. Region 5-6 continues the increasing trend of temperature although the integrated intensity is observed to drop by  $\sim 25\%$ , achieving a local maximum at location 6, corroborated with an increase in temperature. In region 6-7, the emission intensity is observed to decline although the temperature profile is essentially plateaued at  $\sim 3500$  K. Measurements beyond this point resulted in high errors due to low signal and were thresholded at  $\pm 400$  K. The pressure signal is observed to continue its decline, although at 0.17 ms, a spike is observed which is the reflected pressure wave from the initial pressure spike. Such damped reflections were observed over a period of  $\sim 150$   $\mu$ s, approximately the time it takes a sound wave to traverse twice the diameter of the cell. A lack of deviation in the measured temperature suggest the same, as a change in combustion mechanism would have manifested in the measured temperatures. Similar profiles were observed for repeat runs as well. For measuring temperatures at longer durations, where the emission intensity is lower, a complementary test without employing the ND2 filter was done and the result is presented in Figure 2-11.

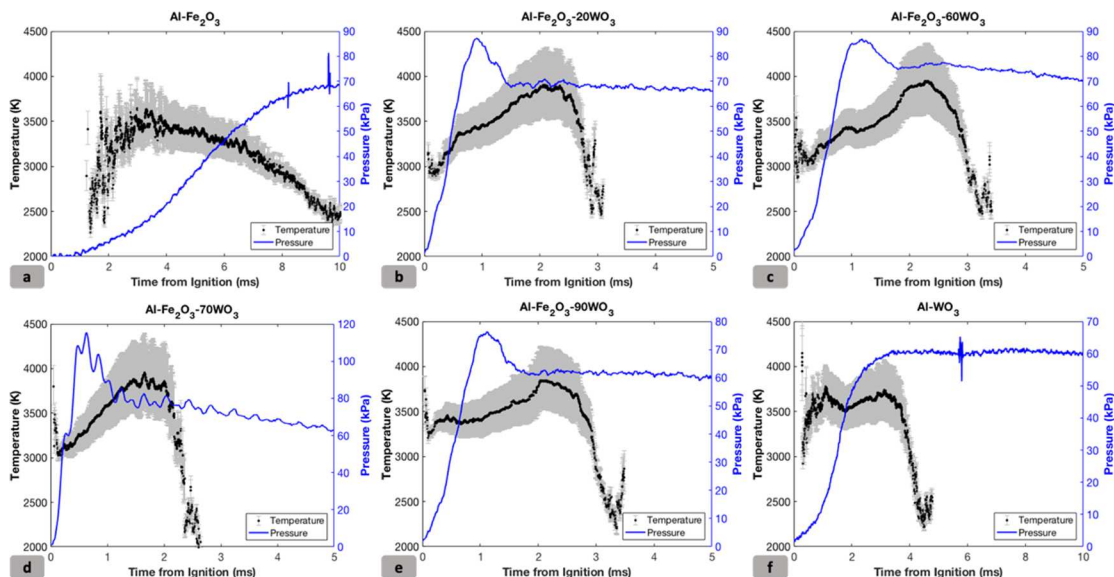
Curiously, the integrated intensity (as shown in section 5-7 of Figure 2-2a) is observed to decline gradually after reaching the maximum, although the recorded

temperature is still observed to increase/plateau. A possible explanation for this observation could be that the flame cloud could have disintegrated into individual emitters which subsequently attenuate the net emission from the reaction due to the reduction in the emission area. Since the lens assembly is focused at the inner edge of the cell, and given that the temperature is seen to rise, it is believed that the spectrometer is observing the combustion of such individual emitters which could be sintered reactants scattered by the initial pressure pulse.

#### 2.4.2 $Al/Fe_2O_3$ and $Al/WO_3$ nanothermite tests in pressure cell

The principal focus of this work is to infer the role of temperature in the tunable reactivity of nanothermites, specifically for  $Al/Fe_2O_3$  nanocomposite doped systematically with  $WO_3$  nanoparticles. The pressure-temperature profiles for select samples are shown in Figure 2-3, and quantitative comparison for the full suite of samples are shown in Figure 2-4. As can be seen in Figure 2-3a for the  $Al/Fe_2O_3$  sample, the pressure profile is characterized by slow buildup which achieves the peak pressure  $\sim 10$  ms after ignition. Such poor performance is again highlighted in Figure 2-4c, where the Full Width Half-Max (FWHM) burn time is plotted against the composition, with  $Al/Fe_2O_3$  being the slowest burning composite at 5.5 ms. With incremental addition of  $WO_3$ , the performance of the composite improves significantly, exemplified by the faster pressure buildup, higher pressurization rate (Figure 2-4b), and much shorter burn time, as shown in Figure 2-4c. With the increase of  $WO_3$  concentration in the composite, the pressure and pressurization rates are observed to increase till it reaches an optimum value at  $\sim 70\%$   $WO_3$  (Figure 2-4b) beyond which any addition of  $WO_3$  resulted in the detriment of the reactivity. For  $Al/WO_3$ , the

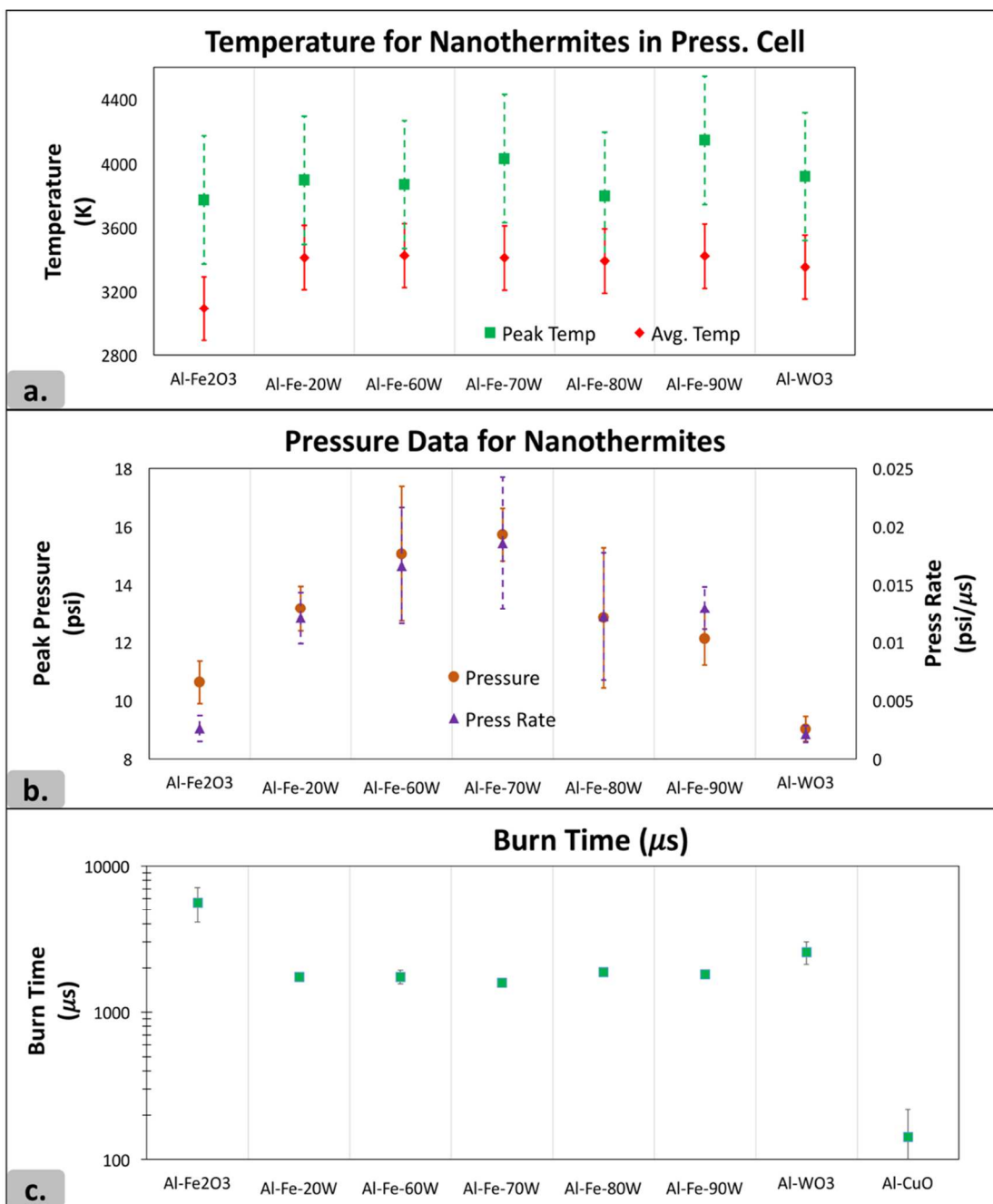
observed pressure buildup was slower than that of the composites with mixed oxides although it is still faster than Al/Fe<sub>2</sub>O<sub>3</sub>, suggestive of higher reactivity as demonstrated by its shorter burn time compared to Al/Fe<sub>2</sub>O<sub>3</sub> (2.5 ms vs 5.5 ms). The lack of any gas phase products for the Al/WO<sub>3</sub> reaction<sup>41</sup> explains its poor pressure metrics among all the composites. The qualitative difference between the respective plots would be analyzed in more detail from a mechanistic standpoint in a later section.



**Figure 2-3:** Temporal Pressure-Temperature profiles for a. Al-Fe<sub>2</sub>O<sub>3</sub>, b. Al-Fe<sub>2</sub>O<sub>3</sub>-20WO<sub>3</sub>, c. Al-Fe<sub>2</sub>O<sub>3</sub>-60WO<sub>3</sub>, d. Al-Fe<sub>2</sub>O<sub>3</sub>-70WO<sub>3</sub>, e. Al-Fe<sub>2</sub>O<sub>3</sub>-90WO<sub>3</sub>, f. Al-WO<sub>3</sub> in pressure cell. The grey region is the error bound of the measurement.

Figure 2-4a highlights the quantitative temperatures measured for the full suite of samples tested in this study (sans Al/CuO). The peak temperatures incorporate a standard error of  $\pm 400$  K, as it was the maximum error allowed in these measurements. With the addition of minimal amounts of WO<sub>3</sub> the average temperatures are seen to rise and plateau at around 3400 K which, as per the previous work by Sullivan et al.<sup>61</sup>, corresponds to the adiabatic flame temperature of Al/WO<sub>3</sub> (3447 K), and slightly higher than the temperature for full Fe<sub>2</sub>O<sub>3</sub> decomposition. The peak temperatures, on the other

hand, is seen to rise gradually with added  $\text{WO}_3$  until it reaches a local maximum at 70%  $\text{WO}_3$ , although the high errors associated with peak temperatures preclude further discussion. Figure 2-4c compares the FWHM burn times for all composites, with Al/ $\text{Fe}_2\text{O}_3$  not surprisingly being the slowest ( $\sim 5.5$  ms) and Al/CuO the fastest ( $\sim 150$   $\mu\text{s}$ ). The other composites show a steady burn time of  $\sim 2$  ms. Al/ $\text{Fe}_2\text{O}_3$ /70 $\text{WO}_3$  had the best performance in the pressure cell with a burn time  $\sim 200$   $\mu\text{s}$  faster, although it is difficult to distinguish on the logarithmic scale. The error bars associated with the burn time measurements were, in some cases, smaller than the marker themselves. Al/CuO pressure data is not shown in Figure 2-4b owing to its large magnitude ( $P_{\text{max}} = 572$  kPa and Press. Rate = 41 kPa/ $\mu\text{s}$ ) which skews the pressure profile, inhibiting visual comparison. All data points presented are averaged over 3 runs.

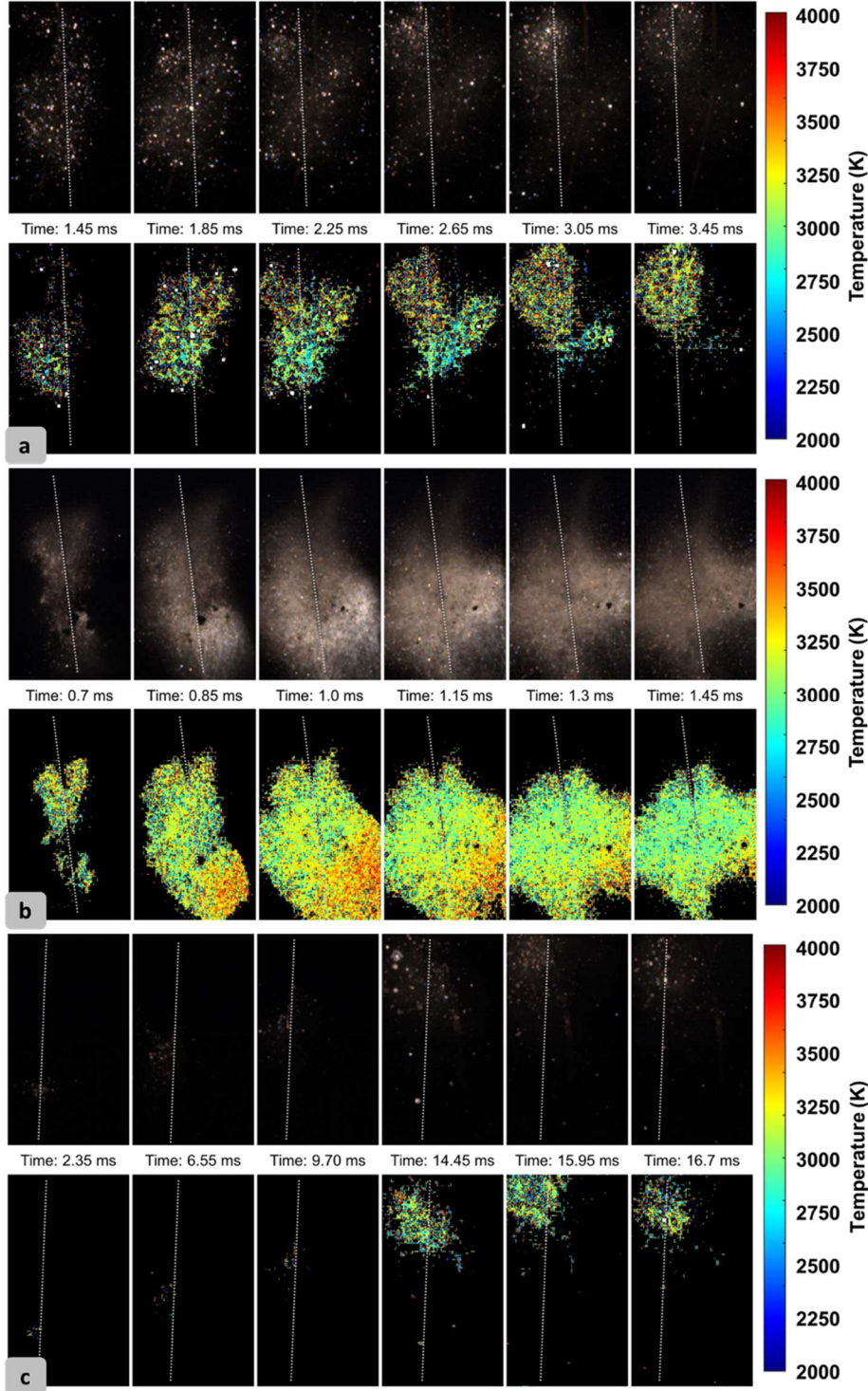


**Figure 2-4:** Pressure-Temperature profiles for a. Effect of composition on temperature, b. Effect of composition on pressure and pressurization rate (Al/CuO:  $P_{\max} = 572$  kPa and Press. Rate = 41 kPa/μs) and c. Effect of composition on Burn time.

#### 2.4.3 Qualitative observation of reaction dynamics with camera

High speed color camera pyrometry videos enabled the identification of the different mechanisms of the Al/Fe<sub>2</sub>O<sub>3</sub>, the Al/Fe<sub>2</sub>O<sub>3</sub>/xWO<sub>3</sub> and Al/WO<sub>3</sub> reactions due

to distinct features that appeared throughout the reaction. For the Al/Fe<sub>2</sub>O<sub>3</sub> mixture, the gas release from the thermite mixture produced a cloud of molten metal and little unreacted material was left as it propagated down the wire (Figure 2-5a). The Al/Fe<sub>2</sub>O<sub>3</sub> reaction is believed to be limited by the oxidizer decomposition and is characterized by the slow reaction of the aluminum with gas phase oxygen released from the oxidizer, as exemplified by its longer burn durations and slow rising pressure profile.<sup>41,61</sup> In comparison, the Al/WO<sub>3</sub> reaction is expected to occur in the condensed phase due to the lack of oxygen release from WO<sub>3</sub>.<sup>41</sup> Upon observing the videos of the Al/WO<sub>3</sub> reaction, the condensed phase nature of this reaction was evident by the absence of a reactive cloud, as the reaction seemingly occurred on the wire and continued on to the ends after melting the wire (Figure 2-5c). In line with the observations from the pressure cell, the Al/Fe<sub>2</sub>O<sub>3</sub>/70%WO<sub>3</sub> sample (Figure 2-5b) shows both higher peak temperature, and a larger combustion zone which was attributed to higher gas release. The use of a color camera to measure temperature allows for the direct observation of the disaggregating role that the oxidizer gas release plays during reaction since the reaction cloud exhibits higher temperatures than the material remaining on the wire.



**Figure 2-5:** High speed pyrometry frames of a. Al/Fe<sub>2</sub>O<sub>3</sub>, b. Al/Fe<sub>2</sub>O<sub>3</sub>/70% WO<sub>3</sub> and c. Al/WO<sub>3</sub> samples ignited on a hot wire at 1 atm. argon environment highlighting the gas production and enhanced reaction. In the Al/WO<sub>3</sub> case, the reaction propagated up the wire over a longer time scale due to lack of gas release from the oxidizer. In each figure, the top image is a gain-adjusted raw image and the bottom image is the 2-D temperature map.



## 2.5 Mechanism

A large body of previous work has been devoted to the mechanistic examination of Al/CuO system and is only briefly described here. It has been shown through high resolution microscopy that the Al/CuO system initiates in the condensed phase<sup>114</sup> where the reactants are postulated to undergo reactive sintering. Based on the observed pressure, emission and temperature profile, the following mechanism is proposed for the Al/CuO system. Upon ignition, the temperature observed in Figure 2-2b is close to the adiabatic flame temperature ( $T_{ad}$ ) of Al/CuO (2837 K)<sup>61</sup> and rises as the pressure wave unloads (region 1-2), with gas phase reactions between aluminum and oxygen, from the decomposition of CuO, dominating the combustion (region 1-2). Due to this spike in temperature, the unreacted copper oxide would undergo endothermic decomposition (as highlighted by the reduction in temperature in region 2-3), producing gaseous oxygen and a rise in pressure. This promotes advection which ignites the bulk sample and continues to burn in the region 3-4, where temperatures measured are close to the  $T_{ad}$ . The gas generation and unreacted material ignition is a symbiotic process and leads to bulk overall combustion as exemplified by the sharp increase in the emission intensity in the region 3-4.

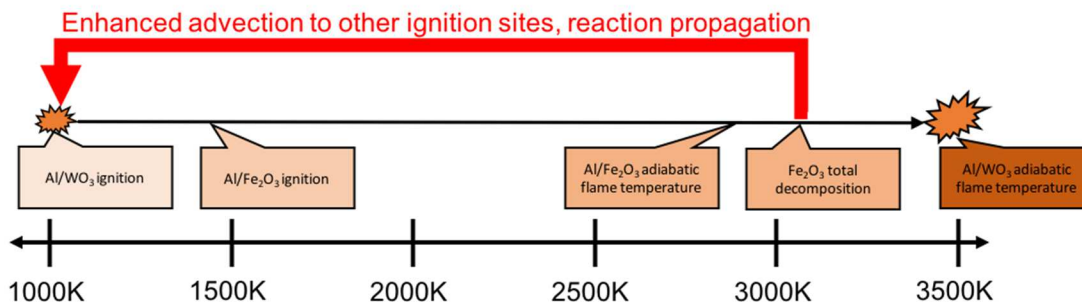
It is believed that the dominant reaction pathway in this region is in the condensed phase through reactive sintering<sup>47,99</sup> as evidenced by large reaction products that are known to form.<sup>47</sup> The rising temperature in region 4-6 results from oxygen released from the decomposition of CuO reacting with Al. The most clear evidence for this is that temperatures seen ~3500 K exceeds the adiabatic flame temperature of Al/CuO and thus implies that aluminum, which has sintered into super-micron

particles, is behaving like aluminum burning in oxygen.<sup>115</sup> Region 6-7 is characterized by a temperature plateau at  $\sim 3500$  K which could be attributed to the combustion of such aggregates, similar to that observed in extended length burn tube tests.<sup>116</sup> The maximum achievable spectral resolution of this instrument is 1 nm/channel (using a 1800 l/mm grating) which is low for resolving the AlO emission band, a common signature of gas phase combustion of aluminum particles, and hence was not adopted in this study.<sup>112</sup>

In comparison to Al/CuO, the Al/Fe<sub>2</sub>O<sub>3</sub> reaction is limited by the slow decomposition of the iron oxide into gas phase oxygen, which leads to prolonged emission traces and slow pressurization rates. The Al/WO<sub>3</sub> reaction is expected to occur entirely in the condensed phase as the WO<sub>3</sub> produces no gas phase decomposition products. Such qualitative differences can be readily seen in Figure 2-3. Beginning with Al/Fe<sub>2</sub>O<sub>3</sub> (Figure 2-3a), upon initiation, the reaction temperature is  $\sim 2400$  K and has a significant delay of  $\sim 1.5$  ms. The temperature is seen to rise gradually over  $\sim 2$  ms, reaching a peak at  $\sim 3500$  K, and dropping gradually thereafter for over 6 ms. The burn time observed for Al/Fe<sub>2</sub>O<sub>3</sub> was the longest of the samples studied with FWHM approx. 6 ms as shown in Figure 2-4c. Gaseous oxygen is initially released by Fe<sub>2</sub>O<sub>3</sub> at approximately 1400 K<sup>61</sup> and the combustion of the nanothermite is believed to be limited by the oxidizer decomposition, since complete decomposition to Fe occurs at ( $\sim 3300$  K).<sup>61</sup>

With addition of WO<sub>3</sub>, the initial reaction temperature of the sample rises by approximately 500 K to  $\sim 3000$  K and is observed for all compositions containing WO<sub>3</sub>. The similarities in temperature is believe to be due to the aluminum initiating with

WO<sub>3</sub>, as it has a lower ignition temperature (1030 K vs. 1410 K),<sup>41</sup> leading to a higher ‘initiation spot’ temperature comparable with that of pure Al/WO<sub>3</sub>. This initial enhancement in temperature could significantly improve the gasification of Fe<sub>2</sub>O<sub>3</sub>, which could help in disaggregating the sample as highlighted by the increase in pressure and pressurization rate in Figure 2-4b. The increase in temperature is observed in equilibrium calculations too, as shown in Sullivan et al.<sup>61</sup> Although the initial temperatures are higher, the concentration of WO<sub>3</sub> in the blend could be too low to influence the entire composite, hence only limited improvement in pressure was noticed. With further addition of WO<sub>3</sub>, the temperature does not change significantly, although the pressure metrics show improvement. The average temperatures in Figure 2-4a are observed to plateau at ~3400 K, close to the adiabatic flame temperature of Al/WO<sub>3</sub> (3447 K) and above the complete decomposition point, shown thermodynamically, of Fe<sub>2</sub>O<sub>3</sub>.<sup>61</sup> The observed temperatures were also higher than the average temperatures observed for Al/Fe<sub>2</sub>O<sub>3</sub> (~3100 K) suggesting heightened oxidizer decomposition. Recent work from the Zachariah Group<sup>39</sup> suggested that nanothermite reactions with a larger amount of gas release tend toward a more complete reaction due to disaggregation of the material, preventing active reactants from coalescing. The enhanced gas release observed in this work for the Al/Fe<sub>2</sub>O<sub>3</sub>/WO<sub>3</sub> composite would suggest that the composite is achieving a higher extent of reaction, resulting in shorter combustion time. Thus, the enhanced gas release could create a feedback loop where the composite disaggregation and convective heat transfer is promoted thereby increasing reactivity, as pictorially represented in Figure 2-6.



**Figure 2-6:** Proposed reaction mechanism.

As suggested in Figure 2-6, ignition of Al/WO<sub>3</sub> (which occurs at 1030 K) brings the mixture to a temperature at which oxygen release in Fe<sub>2</sub>O<sub>3</sub> is initially observed (1400 K).<sup>41</sup> The subsequent disaggregation of material due to oxygen release then contributes to a more complete combustion of the Al/WO<sub>3</sub> that approaches the adiabatic flame temperature of the mixture at 3447 K, at which point the temperature exceeds the total decomposition point of Fe<sub>2</sub>O<sub>3</sub>.<sup>61</sup> The Fe<sub>2</sub>O<sub>3</sub> decomposition into its suboxides (Fe<sub>3</sub>O<sub>4</sub>, FeO, Fe)<sup>60</sup> then releases more gaseous oxygen, leading to further disaggregation of material and propagation of the cycle. This is exemplified by the color camera pyrometry videos in Figure 2-5, where the Al/Fe<sub>2</sub>O<sub>3</sub>/70%WO<sub>3</sub> composites exhibit a larger flame cloud with a higher temperature. Moreover, the reaction is observed to occur away from the wire highlighting the disaggregation effects of gas release.

As the temperatures exceeds full decomposition temperatures of Fe<sub>2</sub>O<sub>3</sub> (~3300 K),<sup>61</sup> a cloud of oxygen gas could be generated in which any residual aluminum, which could have coalesced into larger particle sizes, could now react. The high temperatures approaching ~3800 K, similar to that of micron aluminum combustion in oxygen,<sup>115</sup> could be suggestive of such a mechanism. The resulting figure of temperature and

pressure as a function of oxidizer composition (Figure 2-4) therefore illustrates a complex interplay between  $\text{WO}_3$  content (which elevates the temperature) and  $\text{Fe}_2\text{O}_3$  (which elevates pressure through release of gaseous oxygen). This trend continues until a tipping point near 80%  $\text{WO}_3$ , where the disaggregation through oxygen release from the  $\text{Al}/\text{Fe}_2\text{O}_3$  reaction is not able to influence the bulk of the composite, owing to  $\text{Fe}_2\text{O}_3$  now being the minor component, suggesting a deviation from the optimal composition, in spite of the temperature profile showing similar features. When the composite is entirely  $\text{WO}_3$ , the high temperature regions ( $\sim 3800$  K), observed for the doped composites, are no longer observed and the temperature is close to the adiabatic flame temperature of the  $\text{Al}/\text{WO}_3$  reaction (3447 K).<sup>61</sup>

## **2.6 Conclusions**

The reactivity of metastable intermolecular composites was investigated through high speed spectrometry, pressure measurements, and color camera pyrometry, culminating in a proposed reaction mechanism for tunable thermite reaction. Seven mixtures of  $\text{Al}/\text{Fe}_2\text{O}_3$  were doped with varying amounts of  $\text{WO}_3$  to manipulate the primary reaction mechanism from gas-generating ( $\text{Fe}_2\text{O}_3$ ) to condensed-phase ( $\text{WO}_3$ ). While pressure, pressurization rate and burn time correlate with mixture fraction, temperature was relatively insensitive once a threshold addition of  $\text{WO}_3$  was achieved. Pyrometry videos capture the interplay of reaction mechanisms of the doped thermite mixtures as evidenced by an enlarged reactive cloud size and faster reaction times with increasing amounts of  $\text{WO}_3$  up to the 70% mark. It is proposed that initiation by  $\text{Al}/\text{WO}_3$  reactions leads to a greater degree of reduction of  $\text{Fe}_2\text{O}_3$ . The high oxygen release also results in flame temperatures in excess of the  $\text{Al}/\text{Fe}_2\text{O}_3$  adiabatic flame

temperature and reflects the burning of Al in an oxygen environment. The relative interplay between condensed and gas phase combustion suggests that performance of nanoenergetic materials can be tuned for specific applications by means of complementary reaction mechanisms.

## ***2.7 Acknowledgements***

This work was supported by the Army Research Office. Reproduced from R.J. Jacob\*, D.J. Kline\*, M.R. Zachariah, High speed 2-dimensional temperature measurements of nanothermite composites: Probing thermal vs. Gas generation effects, J. Appl. Phys. 123 (2018) 115902. doi:10.1063/1.5021890 with permission from AIP Publishing.

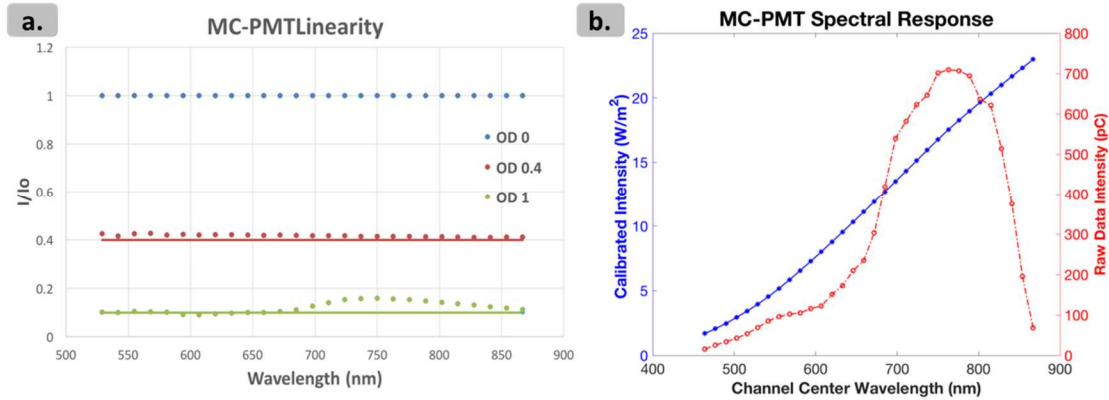
\*Denotes that authors contributed equally to this work.

## ***2.8 Supplemental Information***

### ***2.8.1 Calibration of a high-speed spectrometer***

The linearity of the MC-PMT was tested by illuminating the collector optics with a high temperature black body source at 1273 K and subsequently attenuating the signal using neutral density filters (ND 0.4, ND 1). Neutral Density filters corresponding to ND 0.4 and ND 1 attenuates the broadband light to 40% and 10% respectively. The results of the calibration test are shown in Figure 2-7a for the selected MC-PMT voltage of 600V. As can be seen, the detector is fairly linear although at low light intensities (ND1), due to poor signal to noise ratio, some non-linearity was observed. The wavelength calibration for the spectrometer was done using a HgNe pencil lamp (Newport) and the spectral response calibration was done using a calibrated tungsten halogen lamp (Avantes HAL CAL) operated at 2440 K. Figure 2-7b shows

the RAW counts corresponding to the intensity profile of the calibration lamp, measured at a cathode voltage of 600 V on the MC-PMT. The spectral response of the entire assembly was calculated by placing the calibration lamp at the inlet of the collection lens assembly for obtaining the RAW data counts at the PMT voltage of interest and computing the correction factor by taking the ratio of the RAW counts to that of the calibration curve as shown in Figure 2-7b. Thus, a single correction factor accounts for the detector spectral response, the grating efficiency and the transmission efficiency of the optical train.



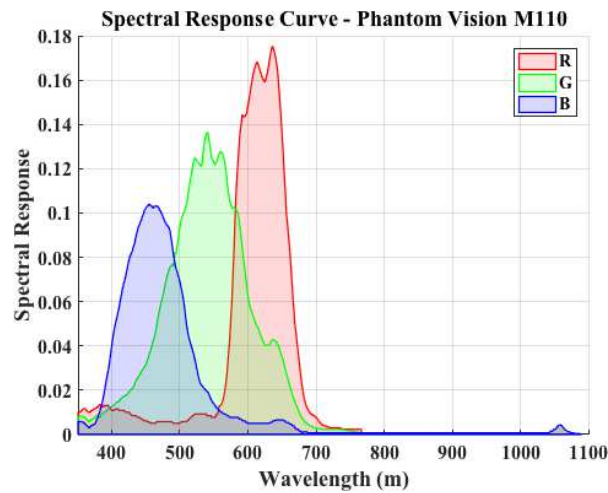
**Figure 2-7:** Calibration for the MC-PMT a. Evaluating the linearity of the detector by attenuating broadband signal using ND filters. Horizontal lines represent expected attenuation. b. Spectral response calibration using a high temperature Tungsten-Halogen lamp.

### 2.8.2 Calibration of a high-speed color camera pyrometer

The intensity of the signal captured by the sensor is dependent on the channel gain, pixel area, solid angle, exposure time, lens transmission, spectral power density of the source, and the spectral transmission of the Bayer filter array integrated over the entire spectrum of sensitivity for the camera.<sup>15</sup> As a result, solving for temperature based on intensity of a single channel becomes inherently difficult as the level of information required for accurate estimations of temperature is prone to error.

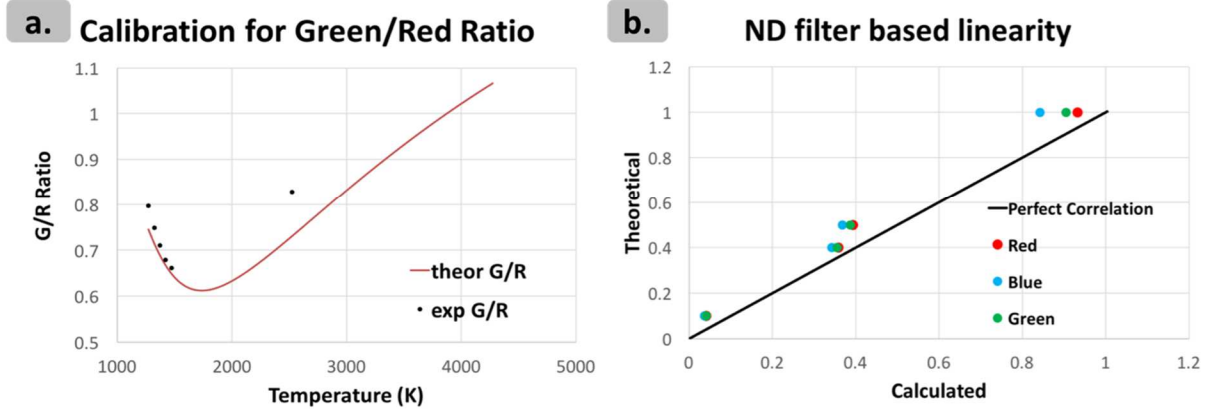
Alternatively, provided that the aforementioned variables (excluding filter sensitivity) remain constant for adjacent pixels of the recorded image, a ratio of the channel intensities recorded by the camera could be used to back-solve for temperature, as shown below, where  $\chi_i$  is the normalized spectral response of the camera and  $\psi_i$  is the correction factor of channel  $i$ . In the case of a gray body radiating source, the emissivity is assumed constant and independent of wavelength and therefore need not be accounted for in calculation, which reduces the measured radiance ( $I_i$ ) ratio to Eq 2-3.

By performing this calculation at different temperatures, theoretical values for intensity ratios between channels can be then used to determine an effective “calibration factor” ( $C_{ij} = \psi_i/\psi_j$ ) for the specific camera components by comparing the expected ratios of intensity to those collected from a standardized temperature source, as shown in Eq. 2-4.



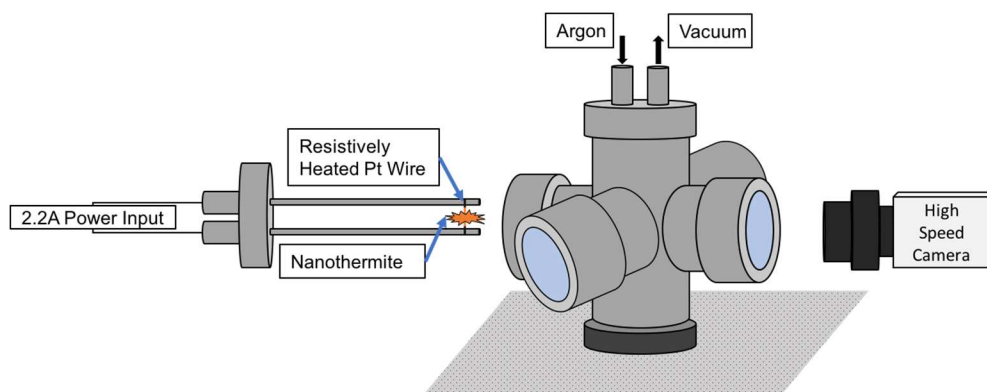
**Figure 2-8:** Spectral response curve of Vision Research Phantom Miro M110 high speed camera as obtained from the manufacturer.



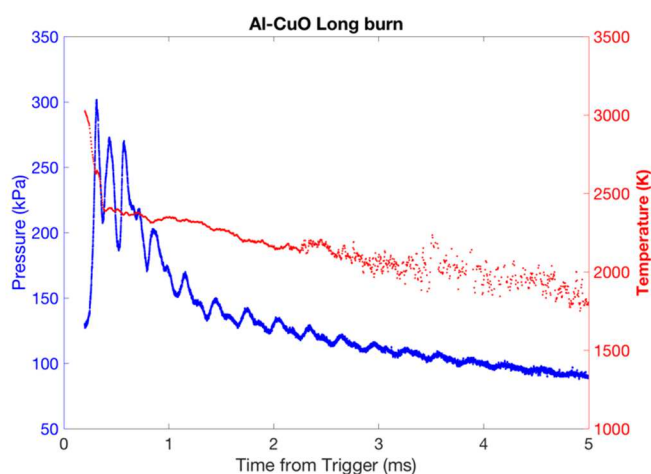


**Figure 2-9:** Calibration for the Color camera pyrometer a. Spectral response calibration using a Black body and high temperature Tungsten-Halogen lamp. b. Evaluating the linearity of the camera sensor by attenuating broadband signal using ND filters. Dark line represents the expected attenuation.

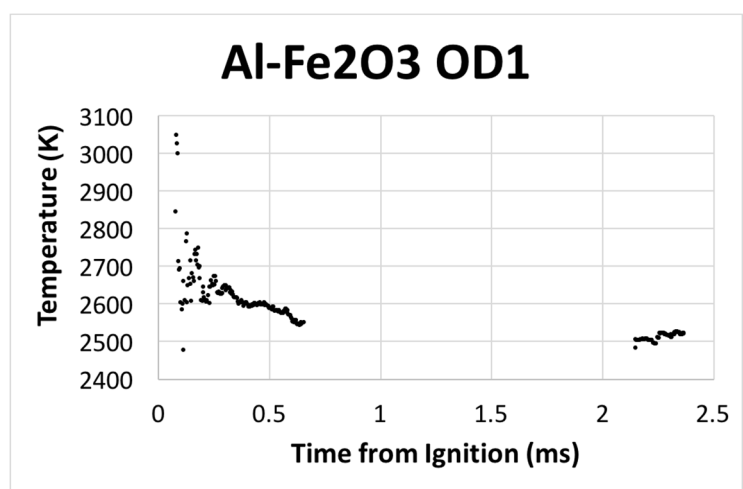
In this work, a high temperature black body (Newport) was used to calibrate the sensor at various temperatures to obtain effective  $C_{ij}$  values. As can be seen in Figure 2-9a, the black body temperature was adjusted between 1273 and 1473 K with 50 K increments. The theoretical estimate from Eq. 2-3 is plotted as the red line with the overlaid black markers being the calculated ratios for the Green to Red channels. The calibration was extended beyond the inflection point by using a high temperature tungsten halogen lamp operated at 2440 K which was factory calibrated. The calibration data was integrated with the spectral response of the camera detector (Figure 2-8) to obtain the experimental G/R ratio. The linearity of the camera sensor was calibrated with various Neutral density filters (ND 0.1, 0.4, 0.5, 1) and the result is plotted in Figure 2-9b along with the expected correlation.



**Figure 2-10:** Experimental for visualizing nanothermite reaction using high speed color camera.



**Figure 2-11:** Temperature measurement of Al/CuO nanothermite in pressure cell over extended durations measured without ND2 filter. The initial parts of the reaction is truncated due to saturated emission on the MC-PMT



**Figure 2-12:** Combustion of Al/Fe<sub>2</sub>O<sub>3</sub> observed using ND1 filter which is used for collecting early emission at the time of ignition. The gap in between is due to saturation of the sensor.

## Chapter 3: Combustion dynamics of nanoaluminum-laden reduced graphene oxide fabricated by laser pulse heating

### **3.1 Summary**

The work investigates the synthesis of high-active content nAl particles from  $\mu\text{Al}$  precursors via laser-induced thermal shock in rGO films. High mass loading rGO/Al sheets with sub-5nm nanoparticles and a near-uniform size distribution were generated with  $\text{CO}_2$  laser pulse widths ranging from 100-500ms. From SEM images, it was observed that diffusion of molten Al through weakened  $\text{Al}_2\text{O}_3$  shells led to formation of nanoparticles at high density through the films, however, sustained high temperatures were believed to reduce the amount of defect sites on the graphene in which particles could form. Thermophysical properties of the rGO/Al composites had a significant impact on the peak temperatures reached by the films during thermal shock and their ability to generate nanoparticles. Combustion tests of the composites in an  $\text{O}_2$ -rich environment showed a  $\sim 70\%$  burn rate enhancement after thermal shock. Loss of nanostructure via reactive sintering is proposed to be the primary limitation in the tunability of the material.

### **3.2 Introduction**

Nanostructured carbon-based allotropes such as graphene and its variety of functionalized forms have demonstrated unique physical characteristics that have led to their widespread integration into chemical<sup>117–119</sup>, physical<sup>120–122</sup>, and biological<sup>123–125</sup> research initiatives. Owing to the highly-crosslinked, 2-dimensional network of  $\text{sp}^2$ -hybridized carbon atoms, graphene-oxide (GO) based materials have a high surface

area, thermal stability, electrical conductivity, and electron mobility which is widely sought after for the next generation of energetic and catalytic solutions.<sup>85,117,126–128</sup> However, since GO is difficult to work with and the material properties are substantially hindered by the overwhelming presence of oxygen, GO is often partially reduced to reduced graphene oxide (rGO) in an attempt to recover some of the material properties of pure graphene and the resulting applications have had major implications on nanoparticle synthesis.<sup>85,129,130</sup>

Multiple studies have demonstrated that carbon allotropes (rGO and carbon nanofibers) can serve as a substrate for in-situ metal nanoparticle formation via either isolation of the metal from its associated salt or formation from a larger primary particle (typically micron-sized).<sup>85,126,131,132</sup> By thermally shocking rGO sheets loaded with a metal or metal salt precursor, unpassivated metal nanoparticles are readily formed in the nano-sized defects of the rGO sheets.<sup>85</sup> Chen et al. successfully produced bare Sn, Si, Au, Pd, and Al nanoparticles by electrically heating micron-sized primary particles (reaching temperatures  $\sim 1700\text{K}$  for  $>10\text{ms}$ ) at low loadings, however the process's dependence on electrical contacts and low loadings does not lend itself to large-scale synthesis.<sup>85</sup>

While these finely tuned particles have promising applications in new electronics devices, they have critical implications in the field of energetic nanomaterial combustion. Decades of research have been dedicated to synthesizing and tuning a new class of energetic materials with heightened reactivities (due to the low diffusion distance between the fuels and oxidizers in the reduction-oxidation reactions of metals) and high specific enthalpies relative to traditional monomolecular

explosives.<sup>6,21,26,38,61,102,133,134</sup> Multiple metals have been investigated as fuels for these energetic nanomaterials, notably aluminum<sup>135</sup>, titanium<sup>55</sup>, and boron.<sup>48</sup> However, while the stability and activity of common oxidizers in these composites is high, the nano-sized fuels have a low active energy content relative to their micron counterparts as a result of the naturally passivating 2-10 nm layer of inert metal-oxide on the exterior of the fuel particles.<sup>42</sup> Studies in the past have attempted to alter the reactivity of nanoenergetic materials by changing fuel particle sizes, however this passivating layer was the primary reason for a plateau in performance.<sup>32,38</sup> Since thermal shock synthesis has successfully generated bare metal nanoparticles (especially Al), integration of these materials could open up new avenues by which to tune reactivity.<sup>85</sup>

While the prospect of bare metal nanoparticles may resolve issues with tuning energetic material performance via fuel particle size for physically mixed samples, Al oxidizes almost instantaneously in the presence of air and isolation of the nanoparticles for integration into other mixtures may prove ineffective or counteractive to enhancing the combustion performance of the material.<sup>42</sup> Therefore, one efficient method for recovery of the heightened Al nanoparticle activity is via combustion of the bulk material akin to a solid propellant.

This paper investigates an alternative, industrially scalable method for ultra-fast, in-situ synthesis of aluminum nanoparticles at high mass loadings in rGO sheets via laser-induced thermal shock. Having generated nanoparticles in the rGO sheets, the combustion performance of the rGO/Al matrix will be investigated as a function of the metal mass loading and the size of the energetic particles.

### **3.3 Methods**

#### ***3.3.1 Preparation of metallized reduced graphene oxide sheets***

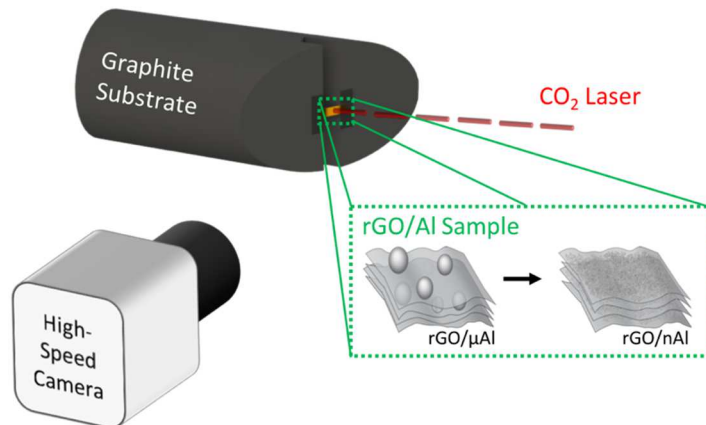
Graphene oxide (GO) solution was prepared via an improved Hummers method and thus is only briefly discussed here. Graphite flakes (325 mesh, Alfa Aesar) are mixed for 12 hours with sulfuric acid, phosphoric acid, and potassium permanganate in a beaker placed in a 50°C bath. The solution is then poured over ice, after which hydrogen peroxide is added and allowed to cool. The solution is centrifuged, separated, and mixed with a hydrochloric acid solution, and centrifuged repeatedly with distilled water to neutralize the solution by removing excess acid and to increase the concentration of the GO. The resulting GO solution is then freeze-dried overnight to remove all water and the remaining material is then resuspended in an ethanol solution at a concentration of 10 mg GO/mL ethanol.

Metallized GO sheets are prepared by extracting 30 mg of GO solution and mixing with the 3.0-4.5  $\mu\text{m}$  diameter Aluminum primary particles (Alfa Aesar). The desired mass of metal is weighed out and added to 1.2 mL ethanol/mg metal. The metal/ethanol solution and GO solution are both mixed and sonicated, added to each other, mixed together, and sonicated. A vacuum filtration system with a 0.4  $\mu\text{m}$ -diameter pore mesh (Millipore) is used to dry the solution and prepare a metal/GO film.

The metal/GO film is thermally reduced to metal/rGO in a high temperature tube furnace by placing the sample between two carbon plates, heating the sample at a rate of 5°C /min, and holding the temperature at 300°C for 120 minutes. The reduction from GO to rGO is carried out in an 5% H<sub>2</sub>/95% N<sub>2</sub> environment to prevent combustion of the sheets at high temperature and inhibit oxidation of the metal particles.

### 3.3.2 Laser-induced thermal shock of metal/rGO sheets

To activate the material for energetic application, a fine dispersion of aluminum nanoparticles accomplished by laser shock heating is needed. To thermally shock the metal/RGO sheets and simultaneously measure temperature of the sample during the heating pulse, a graphite substrate was designed to mount a 4.5 mm x 2.5 mm sample of the metal/RGO film inside a four-way steel cross (see Figure 3-1). A crystalline NaCl window (International Crystal Laboratories), which is transparent to the IR laser, is mounted to the laser inlet of the chamber. The irradiation is absorbed by the film and a high-speed video of the event is captured at a port 90-degrees offset from the laser inlet. The other two ports in the steel cross are used to change the environment and sample. An 80 W laser (Synrad) with a  $2.2 \text{ mm} \pm 0.3 \text{ mm}$  beam diameter is operated at 90% power and is triggered with a house-built pulse-generator box for 100-500 ms. Nanoparticle generation of the sheets is carried out by thermally shocking the material in an Ar environment and combustion experiments with the metallized RGO sheets were performed steady-flow  $\text{O}_2$  environment.



**Figure 3-1:** Apparatus mockup for thermal shock and temperature measurements of rGO/Al films.

### 3.3.3 Temperature measurements

Two-dimensional temperature measurements of the heated films were taken via color-ratio pyrometry on a Vision Research Phantom Miro M110 high-speed camera recording at 3000 frames per second. Dependency on most variables associated with intensity is eliminated except for those regarding the channel gain, emissivity, and spectral response of the camera at individual wavelengths and channels when calculating temperature from raw channel intensities.<sup>15</sup> It was assumed that the material had a constant emissivity (graybody assumption), thus temperature was calculated by substituting this assumption into Planck's Law and integrating over the entire spectrum to which the camera is sensitive (Equation 2-1).<sup>19,136</sup> Calibration factors for  $C_{gr}$ ,  $C_{bg}$ , and  $C_{br}$  were determined in experiments with a Newport Oriel 6700 Series blackbody (data available in Jacob and Kline et al.).<sup>73</sup> With these assumptions and calibration factors, temperature measurements are reduced to matching of calibration factor-corrected channel intensity ratios to theoretical ratios.

A house-built MATLAB was used to extract raw pixel intensities and calculate temperatures. Black-level and saturated pixels were dilated by a factor of 3 and removed from calculations. The demosaicing routine in MATLAB was used with the camera's Bayer color filter array to recover values for red, green, and blue channels at each pixel. Three color ratios (green/red, blue/green, and blue/red) were simultaneously used to estimate temperature by minimizing their summed error from theoretical ratios. Further thresholding was used to eliminate summed errors corresponding to a temperature error greater than 110 K. For the figures that show temperature of a single sample as a function of time, only unsaturated pixels above the black level and within



the error threshold are used to report mean and median temperature of the frame for a contiguous area of at least 10 acceptable pixels.

#### *3.3.4 Particle imaging*

Electron microscopy was used to characterize the materials under investigation and to determine the particle size distribution of the formed nanoparticles. Scanning electron microscopy (SEM) images were captured with a Hitachi SU-70 FEG SEM. Prior to examination under an electron microscope, the surface of the metal/RGO films were exfoliated with tape to expose the interior layers of the material. ImageJ was used to determine the particle size distribution of the materials formed in the sheets.<sup>137</sup>

#### *3.3.5 Thermogravimetric analysis/differential scanning calorimetry*

Thermogravimetric analysis/differential scanning calorimetry (TGA-DSC) were used to preliminarily investigate the impact of thermal reduction on the total mass of graphene oxide and to characterize the primary micron aluminum particles. In the case of probing mass loss during thermal reduction of GO to rGO, the procedure described for the reduction process above was repeated in an Ar environment. To determine active Al content in the primary particles, TGA-DSC was carried out in an O<sub>2</sub> environment to fully oxidize the particle.

### **3.4 Results and discussion**

#### *3.4.1 Characterization of original rGO/Al films*

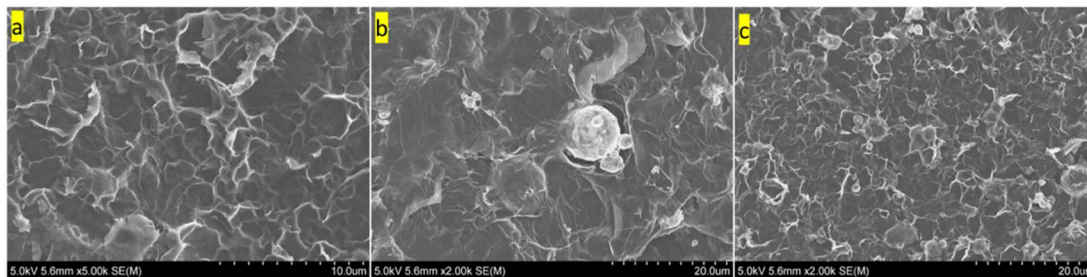
Prior to shock-heating the metallized rGO films, experiments were performed on the original sample containing micron-sized aluminum particles at various mass

loadings. It was assumed that the primary  $\mu\text{Al}$  particles have a 2-10 nm thick, naturally passivating  $\text{Al}_2\text{O}_3$  shell on the exterior which is formed in the presence of an  $\text{O}_2$ -rich environment and the active-Al content of the particles would be  $>99\%$ .

A simulated thermal reduction on a GO film was carried out with TGA-DSC and demonstrated a peak mass loss of  $\sim 58\%$  (see Figure 3-8), thus increasing the rGO/Al mass ratios as they were prepared (see Table 3-1). SEM images of the as-prepared films showed that the original films had clusters of 3-5 primary particles evenly distributed in the samples (see Figure 3-2).

**Table 3-1:** Sample  $\mu\text{Al}$  mass loadings as prepared (left) and after GO reduction (right).

GO/Al Sample as prepared ( $m_{\text{GO}}:m_{\mu\text{Al}}$ )	rGO/Al Sample ( $m_{\text{rGO}}:m_{\mu\text{Al}}$ )
10:1	4.2:1
5:1	2.1:1
2:1	0.8:1



**Figure 3-2:** SEM images of the original rGO/ $\mu\text{Al}$  sheets for mass loading ratios of (a) 4.2:1, (b) 2.1:1, and (c) 0.8:1.

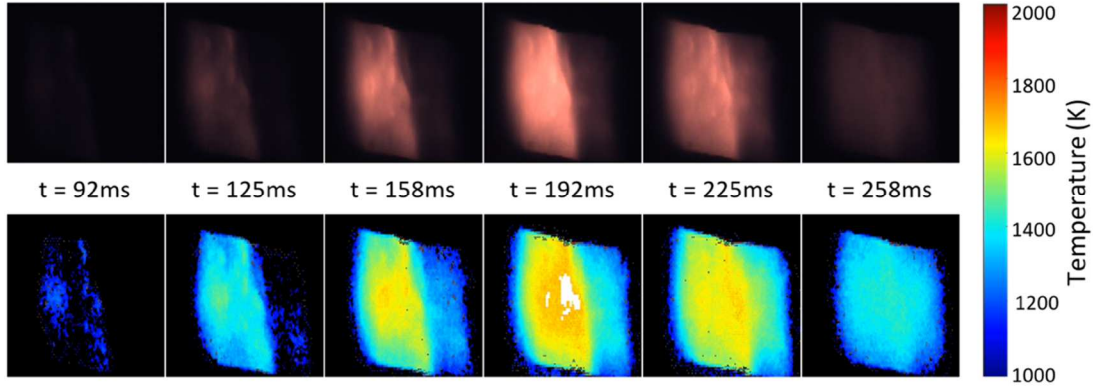
Since Al has a significantly higher enthalpy of combustion when compared to the carbon framework that encapsulates it,<sup>21</sup> the change in mass ratio also increases the theoretical energy release of the composite and its potential as an energetic material with an improved energy density. However, despite the desired increase in energy density, the increase of the average specific heat capacity of the film with the addition of Al also makes thermally shocking the film more difficult with a fixed power source.

### 3.4.2 *In-situ nanoparticle generation*

Provided the relatively large content of active Al in the rGO sheets, thermal shocking of the metallized films in an inert environment was expected to produce high-active content nanoparticles relative to those that are commercially purchased. Considering the relative impermeability of stacked-layer GO and rGO systems to gases due to their large electron clouds and small pore defect sizes<sup>138–141</sup>, the interior particles are likely to retain the original active Al content (making them effectively bare nanoparticles).

High-speed two-dimensional pyrometry of the thermal shock event (see Figure 3-3) showed the entire surface of the rGO/Al sheet begins to heat rapidly around a hot spot at the focal point of the laser. The temperature of the film increases around this hot spot to eventually have a relatively homogeneous temperature field. Peak temperatures at point locations of the films are >1500 K for all of the samples with pulse widths  $\geq 250$  ms, however, the median temperatures of the samples are inversely proportional to the Al mass loading of the film. A brief investigation of the thermal properties of the material reveals an inherent limitation of using a fixed power source for thermal shock of the rGO/Al films since the bulk thermal conductivity ( $k$ ) and specific heat capacity ( $C_p$ ) of the films increases with Al content.<sup>142</sup> With a higher specific heat capacity, the ability for the material to heat up at the same rate (or attain the same temperature) as lower Al-loading samples with the fixed power source is significantly diminished. The peak temperature attainable by the film is even further reduced when considering the impact of the increased thermal mass coupled with the increase in thermal conduction losses to the substrate/film. Considering these bulk

thermophysical material changes, thermally shocking rGO/Al films uniformly to temperatures  $>1600$  K is generally limited by the fixed-power laser used in this experiment. In comparison to the experiments conducted by Chen et al., the power input to the sample by the laser is significantly less than that delivered by the electrical power source. Chen et al. capitalizes on the high electrical conductivity of the film and their heating time is shorter than that demonstrated here since this method is limited by the optical properties of the film at IR wavelengths.<sup>85,143</sup> Despite this, the peak temperatures of the samples between the works is comparable.



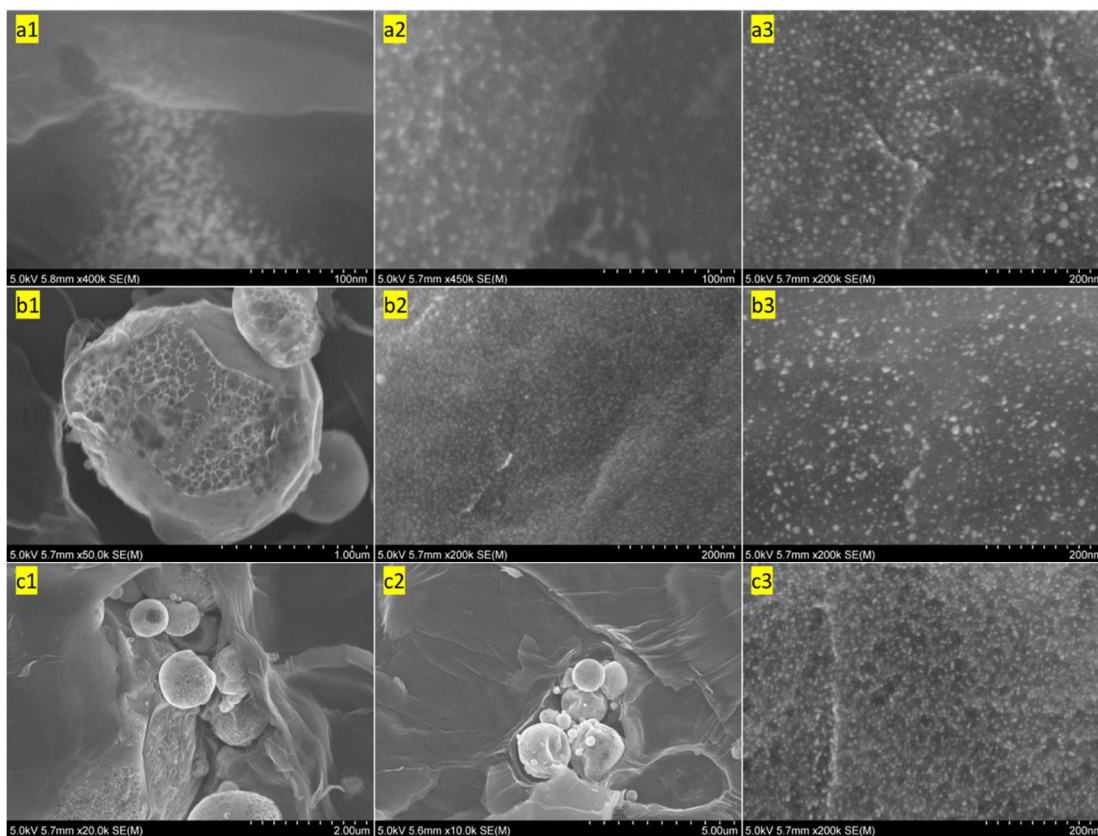
**Figure 3-3:** High speed pyrometry frames of rGO/Al 2.1:1 sheet being thermally shocked with an 80 W CO<sub>2</sub> laser (90% power, 2.2 mm beam diameter) for 250ms in an Ar environment to produce nanoparticles. White areas represent pixels that were saturated. The top image is a gain-adjusted raw image and the bottom is the 2-D temperature map. Sample size is approximately 4.5 mm x 2.5 mm.

Scanning electron microscopy (SEM) of metallized rGO films before and after thermal shock demonstrate the importance of the sustained high temperatures and rapid quenching on the in-situ generation of nanoparticles (see Figure 3-4 and Figure 3-9). Sub-5 nm Al nanoparticles could be found uniformly dispersed around the rGO/Al sheets for the thermally shocked samples that reached local temperatures  $>1600$  K and median film temperatures  $\geq 1450$  K. Upon exfoliation of the thermally shocked rGO/Al sheets, a higher density of sub-5nm nanoparticles could be seen uniformly dispersed

throughout the films (Figure 3-4). For the lower loading samples which were heated for longer times, particle size appears to increase. This is likely attributable to the reduction in defect concentration of rGO at high temperatures<sup>142</sup> and reveals an interplay between the defect size/concentration and the nanoparticle size/distribution as a function of the heating time. For the particles that are generated within the sheets, it is believed that the particles are either bare or have a very small (sub-1 nm) passivating oxide layer due to the high active Al content of the original particles, the relative impermeability of O<sub>2</sub> to rGO, thermal shocking of the samples in an inert environment, and the relatively low amount of oxygen remaining in the rGO matrix.<sup>138,142</sup>

The mechanism of nanoparticle generation, as previously proposed by Chen et al., suggests that the rapid thermal pulse melts the Al precursor particles (~933 K) which wets the interstices of the rGO sheets.<sup>85</sup> The sustained temperature enables Al to diffuse through the Al<sub>2</sub>O<sub>3</sub> shells. In the case of Chen et al., shutting the power input results in a very rapid quench that creates ultrasmall Al nanoparticles stabilized by defects in the rGO sheets. Any vaporized Al within the sheets is likely to remain within the rGO matrix due to its low permeability. A closer look at an rGO/Al 4.2:1 sheet shocked for 100 ms (Figure 3-4a1) served as a basis for the confirmation of this mechanism with small nanoparticle-laden regions being found directly around partially-fractured Al<sub>2</sub>O<sub>3</sub> shells from the precursor particles. These images suggest that the process involves the diffusion of molten metal out of defects in the amorphous shell which then form nanoparticles upon quenching. Other images of thermally shocked films that did not generate nanoparticles (Figure 3-4b1, c1, and c2) also have evidence

of partially-fractured  $\text{Al}_2\text{O}_3$  shells with raised ligand structures that may be reminiscent of Al that began to diffuse out of the shells. Compared to the analogous nanoparticle generation mechanism expected for molecular precursors in the same material configuration, the time scale required to break bonds of small molecules should be significantly shorter than that required for the diffusion-limited process with significant thermophysical contributions from the large primary particles.



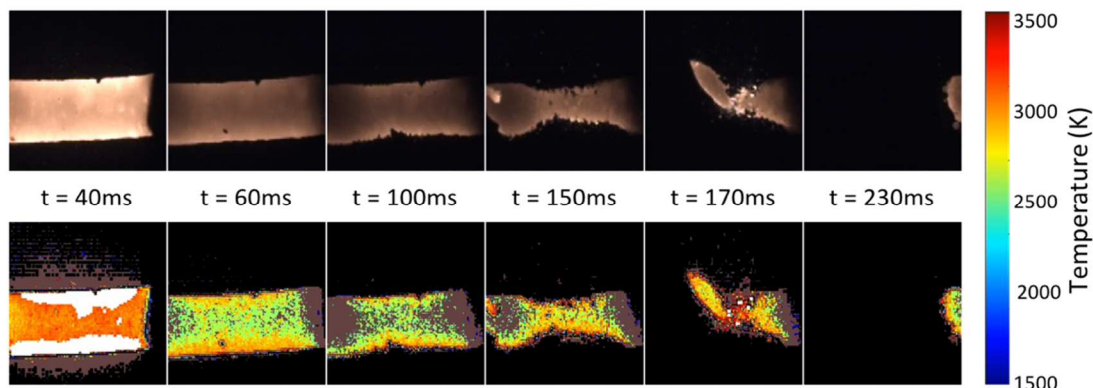
**Figure 3-4:** SEM images of rGO/Al films after thermal shock with a 32.7 W/mm  $\text{CO}_2$  laser in an Ar environment. rGO/Al mass loadings/heating times are (a1) 4.2:1/100 ms (nanoparticles diffusing out of a shell), (a2) 4.2:1/250 ms, (a3) 4.2:1/500 ms, (b1) 2.1:1/100 ms, (b2) 2.1:1/250 ms, (b3) 2.1:1/500 ms, (c1) 0.8:1/100 ms, (c2) 0.8:1/250 ms, and (c3) 0.8:1/500 ms. For each experiment, the primary particle size was 3-4.5  $\mu\text{m}$  diameter Al. All images shown were taken from the first 1-2 layers below the surface of the rGO/Al sheet.

### 3.4.3 Combustion characteristics of rGO/Al films

Combustion of the rGO/Al films were carried out for both micron- and nano-sized Al particles that were generated via the laser induced thermal shock. With a relatively high thermal stability of rGO in air and low O<sub>2</sub> content relative to the Al content, a flow of O<sub>2</sub> was introduced to the chamber depicted in Figure 3-1 (kept constant across all experiments) to ensure adequate oxidizer concentrations.

Upon laser ignition, a two-stage combustion process was evident: (1) ignition of the surface nanoparticles and (2) subsequent exfoliation of the rGO/Al film with sustained combustion of Al in O<sub>2</sub> (Figure 3-5). Within a few milliseconds of powering the laser, a bright flash is easily observable over the entire surface with measured temperatures exceeding 3200 K. As the surface particles oxidize, the high sustained temperatures begin to decompose the underlying rGO sheets (which have an onset of decomposition ~725 K in O<sub>2</sub>, Figure S3) into gaseous compounds which breaks apart the films. The decomposition of the films increases the reacting surface area and the number of particles exposed to the O<sub>2</sub> atmosphere on all sides of the material. The continual combustion of rGO in O<sub>2</sub> (adiabatic flame temperature ~3285 K)<sup>9</sup> exceeds the melting temperature of Al (~933 K), thus the metal inside the sheets is also able to contribute to the heat of reaction. The higher temperatures that are seen towards the flame front in the material is likely a combination of the Al/O<sub>2</sub> reaction (adiabatic flame temperature ~4084 K)<sup>9</sup> and the rGO combustion, which had an estimated adiabatic flame temperature of 3569 K for the mixture.<sup>9</sup> With gradual weakening of the films from the high temperatures and gas liberation, the film typically breaks in the center

point during combustion experiments and the reaction continues to propagate down the length of the film at nearly constant temperature.



**Figure 3-5:** High speed pyrometry frames of rGO/Al 2.1:1 combusting in an O<sub>2</sub> flow (after a 250 ms thermal shock in Ar to generate nanoparticles). White areas represent pixels that were saturated and brown areas are high in error. The top image is a gain-adjusted raw image and the bottom is the 2-D temperature map. Sample size is approximately 4.5 mm x 2.5 mm.

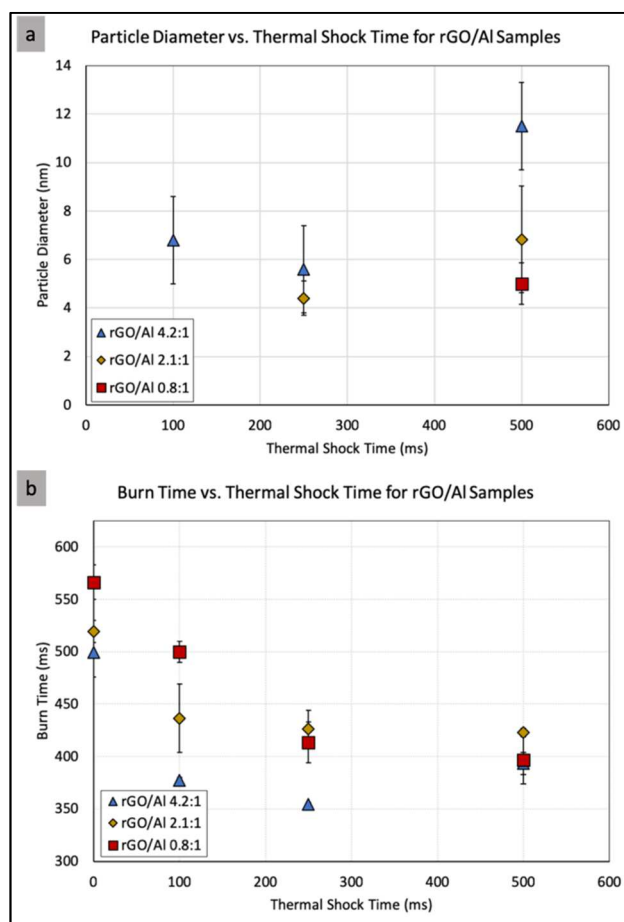
Figure 3-6 shows the effect of how nanoparticles are generated and nanoparticle loading as a function of burn rate. Figure 3-6a shows that the laser shock heating process can make very fine Al nanoparticles with particle sizes less than 10 nm. Another important observation is that the length of the laser pulse appears to be a relatively weak variable in tuning nanoparticle size. Considering that sustained high temperatures of the rGO can lead to defect merging in the sheets and that particles are formed in the defects,<sup>85</sup> the most significant factor in nanoparticle formation is likely defect size and concentration. The impact of sustained temperature on defect concentration is particularly important for the high-loading Al samples since the rGO has a lower thermal mass than Al. Addition of Al particles increases the overall thermal mass of the sample which lowers peak temperatures and sustains defect size as Al content increases (for the same pulse width). The apparent effect of Al loading on



particle size likely reflects that high loadings reduce peak processing temperature and leads to larger particles. Figure 3-6b shows the effect of laser pulse on combustion behavior, indicating that pulse length does not affect particle size significantly and has little effect on burn time.

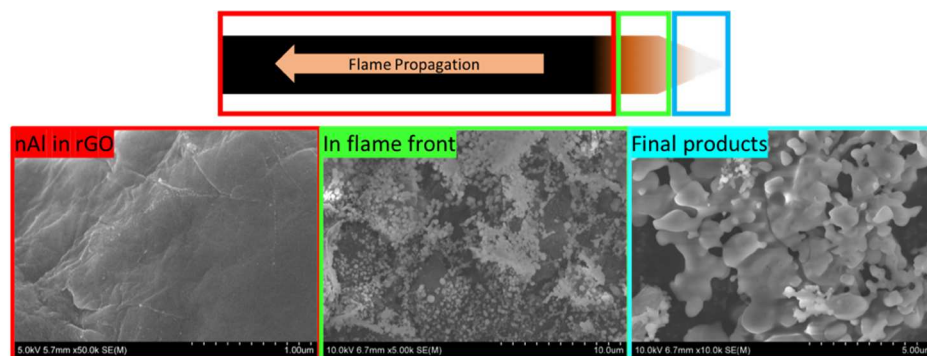
To explore this behavior further, post-combustion analysis by SEM is shown in Figure 3-7 as a function of location relative to the flame front. The nanoparticles become more mobile and melt as they are subject to increasing temperatures near the flame front. Due to mass transfer limits, combustion likely takes place after liquification and coalescence of the molten aluminum, resulting in the burning of larger aluminum domains. This is corroborated by the SEM images of the post-combustion region which shows irregularly shaped particles several microns in diameter. Ironically, this is the initial size of material before the laser shock generation of nanoparticles.

This loss of nanostructure is an inherent limitation of the proposed combustion in rGO/nAl sheets at high loadings with an enhanced and tunable reactivity since the nanostructure should reduce the diffusion distance between reactants.<sup>21</sup> This presumably explains why creating smaller particles has little benefit at the higher loadings where coalescence is facilitated, therefore the observed result is actually combustion of large particles.



**Figure 3-6:** (a) Average particle diameter of Al in rGO after thermal shock for different loadings. For data points not shown, particles were the  $\sim 3\text{-}4.5\ \mu\text{m}$  primary Al particles. (b) Burn time for  $4.5\ \text{mm} \times 2.5\ \text{mm}$  films versus laser-induced thermal shock time of the rGO/Al films at different loadings.

Despite the limitation of the material due to coalescence, it is important to note that the nanoparticle-laden rGO sheets that were thermally shocked consistently burned to completion and propagated in the absence of laser power while the rGO/ $\mu\text{Al}$  counterparts only did so inconsistently. This is believed to be resultant of the thermal shock essentially redistributing the Al more homogeneously through the matrix.



**Figure 3-7:** Images representing the Al particle and rGO sheet morphology as a function of position relative to the flame front during the combustion process. Degree of coalescence in Al in the products region demonstrates the impact of reactive sintering on the combustion process. Images were taken from a sample of rGO/Al 2.1:1.

Considering the effect that thermal shock and nanoparticle generation has on the enhanced dispersion of high-active content Al in the films, this preparation method sets the stage for a “switchable” energetic material which can be readily prepared and stored for long periods of time until it is necessary to use. In one scenario, it could be envisioned that a facility could scale up the production of rGO with homogeneously dispersed  $\mu\text{Al}$  embedded, cast, or printed into various geometries, and then “activate” the material with a radiative thermal shock to better disperse Al within the device prior to ignition.

### 3.5 Conclusions

The synthesis of high-active content nAl from  $\mu\text{Al}$  precursors via laser-induced thermal shock in rGO films was investigated at high mass loadings via color camera pyrometry and scanning electron microscopy, revealing the impact of thermophysical properties of the individual components on their relative capability to generate nanoparticles with a fixed power source. Sub-5 nm Al nanoparticles with a near-uniform size distribution were generated for mass loadings ranging from 4.2-0.8:1 rGO/Al after heating pulses from a  $\text{CO}_2$  laser with pulse widths ranging from 100-500

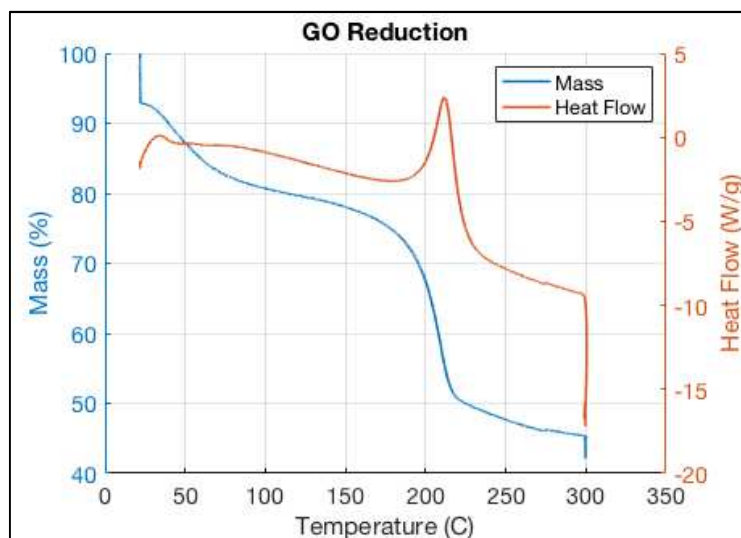
ms. The combustion performance in an O<sub>2</sub>-rich environment revealed significant changes in burning speed after nanoparticle generation and burn rate enhancements up to 70%. SEM images of the products suggest a significant loss of nanostructure in the Al particles due to reactive sintering at high temperatures, thus suggesting that a plateau in reactivity of sheets thermally shocked for longer activation pulse widths is likely due to the degree of homogeneous particle dispersion within the rGO/Al film.

### 3.6 Acknowledgements

This work was supported by the National Science Foundation award #1635221 (SNM: Continuous Synthesis and Stabilization of Nanoparticles in a Carbon Matrix Using Rapid Thermal Shock).

### 3.7 Supplemental Information

#### 3.7.1 Characterization of original rGO/Al films



**Figure 3-8:** TGA-DSC results of mass loss of GO during reduction. Sample was heated at 5°C /minute and held at 600°C for 120 minutes. Final mass was 42% of original.

Ideally, better dispersion of the particles would further reduce an inhomogeneities in the film and prevent heat sinks within the sample during thermal shock, however the processing improvements for the films with a larger vacuum filtration pump and solvent change to ethanol already significantly increased the homogeneity and production scalability over older methods. It is important to note that the films become more fragile with increasing Al content since the rGO is integrated as a binder for the particles.

### 3.7.2 In-situ nanoparticle generation for different loadings

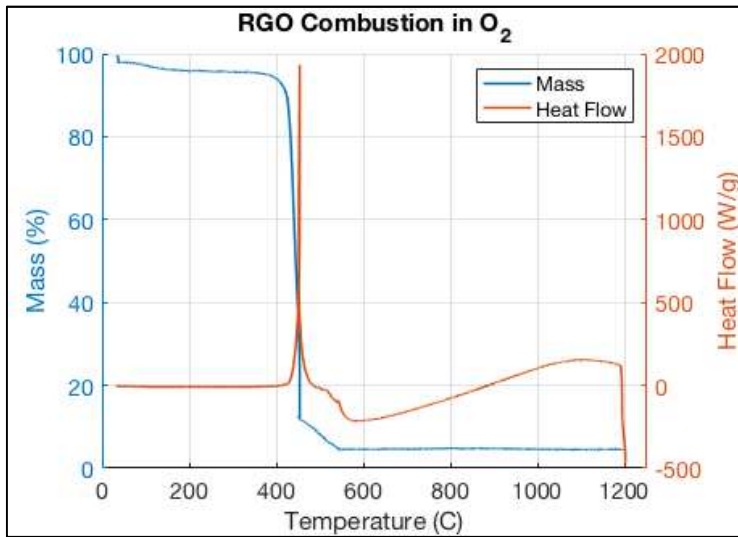


**Figure 3-9:** Sample image of the exterior surface particles for a rGO/Al 4.2:1 film heated for 250 ms. The particles formed have larger particle sizes with a wider distribution in comparison to those found in the interior 1-2 layers (as shown in Figure 3-4).

Initial inspection of the rGO/Al film surfaces revealed the uniform generation of nanoparticles across the entire surface of thermally shocked samples with a wide size distribution and particle diameters ranging from 5-50 nm (see Figure 3-9). Furthermore, in the cases with clear nanoparticle generation, particle morphology on the exterior layers did not vary widely as a function of the original Al loading. In

comparison to the surface layer nanoparticles, those that formed on the interior layers of the rGO/Al films consistently had smaller particle diameters, narrower size distributions, and higher nanoparticle likely due to a higher defect concentration and a reduction in Al evaporation.

### 3.7.3 Combustion characteristics of rGO/Al films



**Figure 3-10:** TGA-DSC results of RGO heated in an O<sub>2</sub> environment. Sample was heated at 25°C /minute to 1200°C. Final mass was ~4% of original.

## Chapter 4: *In-operando* high-speed microscopy and thermometry of reaction propagation and sintering in a nanocomposite

### 4.1 Summary

An important proposed mechanism in nanothermites reactions – reactive sintering – plays a significant role on the combustion performance of nanothermites by rapidly melting and coalescing aggregated metal nanoparticles, which increases the initial size of the reacting composite powders before burning. Here a high-speed microscopy/thermometry capability that enables  $\sim \mu\text{s}$  time and  $\sim \mu\text{m}$  spatial resolution is demonstrated and applied to image highly exothermic reaction propagation. This enabled direct observation of reactive sintering and the reaction front at high spatial and temporal resolution. Experiments on the Al + CuO nanocomposite system reveals a reaction front thickness to be  $\sim 30 \mu\text{m}$  and the temperatures in excess of 3000 K, resulting in a thermal gradient in excess of  $10^7 \text{ K m}^{-1}$ . The local microscopic reactive sintering velocity is found to be an order of magnitude higher than macroscale flame velocity. In this observed mechanism, propagation is very similar to the general concept of laminar gas reaction theory in which reaction front velocity  $\sim (\text{thermal diffusivity} \times \text{reaction rate})^{1/2}$ .

### 4.2 Introduction

Compared to conventional CHNO explosives such as TNT, RDX, and HMX, energetic nanocomposites such as nanothermites are attracting more attention for their

high enthalpy of reaction and environmentally benign products.<sup>144–150</sup> Their high surface area gives them a substantially higher energy release rate in comparison to their micron counterparts, thus making them potential candidates for materials whose reactivity fall in between primary explosives and conventional pyrotechnics. This unique category, which has both military and civilian applications, are limited in widespread implementation due to the inherent complexity of heterogeneous combustion that has yet to be fully understood. For example, theoretically, when reducing the size of the composition constituents to the nanoscale, the energy release rate should be enhanced by orders of magnitudes due to a highly increased interfacial area and reduced diffusion distance between fuel and oxidizer,<sup>43,145</sup> however, they have yet to exhibit such impressive enhancements.<sup>38,46,151</sup> One of the major concerns that may explain this underwhelming enhancement is the loss of nanostructure during reaction.

Sullivan et al.<sup>47</sup> found that an important component in the combustion of nanothermites – reactive sintering – plays a significant role by rapidly melting and coalescing aggregated aluminum nanoparticles and increases the initial size of the reacting composite powders before burning (sintering time  $\leq$  reaction time). Further studies by Chakraborty et al.<sup>58</sup> conducted a reactive molecular dynamics simulation and found that the sintering behavior was motivated by an induced built-in electric field. The loss of nanostructure due to reactive sintering in nanothermite powders dramatically impacts the combustion of the composite and, as a result, burn times of the materials do not significantly shorten as a function of diameter and small particle sizes.<sup>38,46,151</sup> Other studies have shown that the fractional scaling law observed for the



gas phase burning of nanoparticles can be corrected for by considering sintering effects prior to burning<sup>152</sup>. Egan et al.<sup>153</sup> used dynamic transmission electron microscopy to observe morphological changes in nanoparticle aggregates occurring and found that the coarsening process initiates within 15 ns and is completed in less than 50 ns, however, the heating in this case was by a laser and thermometry was unavailable. Even though the reactive sintering phenomenon was speculated in a motionless heating stage TEM and was simulated by the related models, the direct observation of a dynamic reactive sintering process and subsequent propagation process in a nanothermite has not yet been observed in realistic operating conditions.

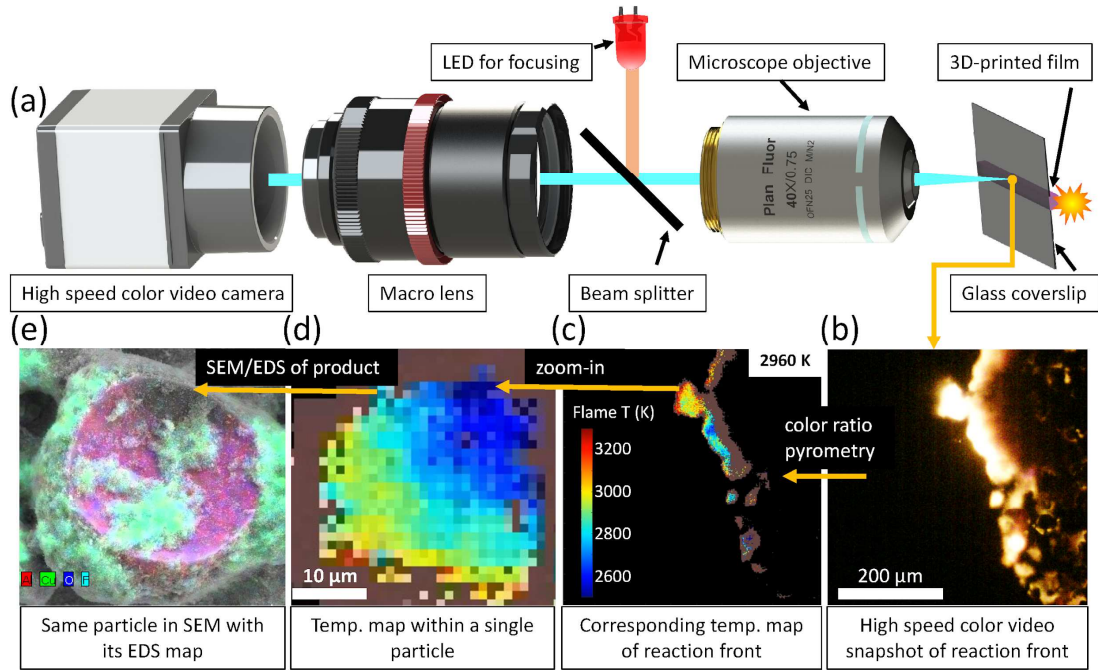
Numerous studies have been conducted with various approaches to directly observe the reaction dynamics of the nanothermites, but none have been able to practically probe a propagating reaction front or reactive sintering on a time or length scale commensurate with the phenomena. Some macroscale studies of thermite reaction dynamics have been captured by T-Jump/time-of-flight mass spectrometer (TOFMS) developed by Zhou et al.<sup>79</sup>, which rapidly heats ( $\sim 10^5 \text{ K s}^{-1}$ ) a material and can provide time resolve-ignition and reaction products profiles. Another commonly employed device to quantify reaction dynamics in bulk materials has been a specially modified constant volume combustion cell developed by Sullivan et al.<sup>61</sup> to simultaneously measure the pressure and optical emission histories of a nanothermite reaction. However, while these measurements provide useful insight on the energy release rate and reaction mechanisms in materials on the appropriate reaction time scales ( $\sim \mu\text{s}$ ), they are unable to capture the dynamics and observe the reaction at the particle length scale ( $\mu\text{m}/\text{nm}$ ). Attempts to directly observe the reaction at these length

scales have been performed using fast-heating stages for scanning electron microscopy (SEM) or transmission electron microscopy (TEM), which did show sintering before and after the heating, however was unable to resolve the event on a nanothermite reaction time scale ( $\sim\mu\text{s}$ ).<sup>153–155</sup> To date, a fast-response, in-operando technique has yet to be employed to observe nanothermite reactions at resolutions high enough to resolve particle-sized phenomena and on the reaction time scale ( $\sim\mu\text{s}$ ).

In this paper, a technique is introduced which bridges the gap between in-operando fast response ( $\mu\text{s}$ ) and high resolution ( $\mu\text{m}$ ). In this technique, a microscope objective coupled to a high-speed color video camera enables observation of the in-operando microscale reaction and propagation of nanocomposite thermites. The high-speed video ( $\sim 55 \mu\text{s}$  per frame) of flame propagation and reactive sintering were captured within a  $512 \mu\text{m} \times 512 \mu\text{m}$  zone of reacting material with a resolution of 1 pixel:  $1 \mu\text{m}$ . Furthermore, the color video could through appropriate calibration and using the RGB filter within the camera, enable 2-D pyrometry temporal temperature maps of the reaction zone. Having imaged the reaction, the exact zone and the corresponding sintered particles were found in a scanning electron microscope (SEM) and matched perfectly with the particles found in the microscope. By employing these techniques, images captured demonstrate reaction propagation in nanothermites and reactive sintering on the microscale in both space and time.

## 4.3 Results

### 4.3.1 Evaluating the reaction zone



**Figure 4-1:** (a) Schematic showing high-speed microscope imaging of 3D-printed reactive materials. Read from right to left: (b) High-speed color video snapshot of reaction front; (c) Corresponding temperature map of reaction front; (d) Temperature map within a single particle, and (e) the same particle in SEM with its EDS map. CAD file of microscope objective courtesy of ThorLabs.

Figure 4-1a shows the major configuration of this study, in which a key point is employing a 40x microscope objective coupling with a high-speed video camera. The microscope objective was focused on the backside of a cover glass slide on which the sample film was printed, thereby allowing the visualization of the flame front without the generated products obscuring the view. With the microscope objective, the pixel/size ratio was  $\sim 1 \mu\text{m}/\text{pixel}$ . At this resolution, the flame front (Figure 4-1b and Figure 4-1c), as well as a single sintering particle (Figure 4-1d and Figure 4-1e), could be captured and the corresponding temperature map (Figure 4-1c and Figure 4-1d) could be obtained. The area was labeled by a thin marker before the ignition, thus

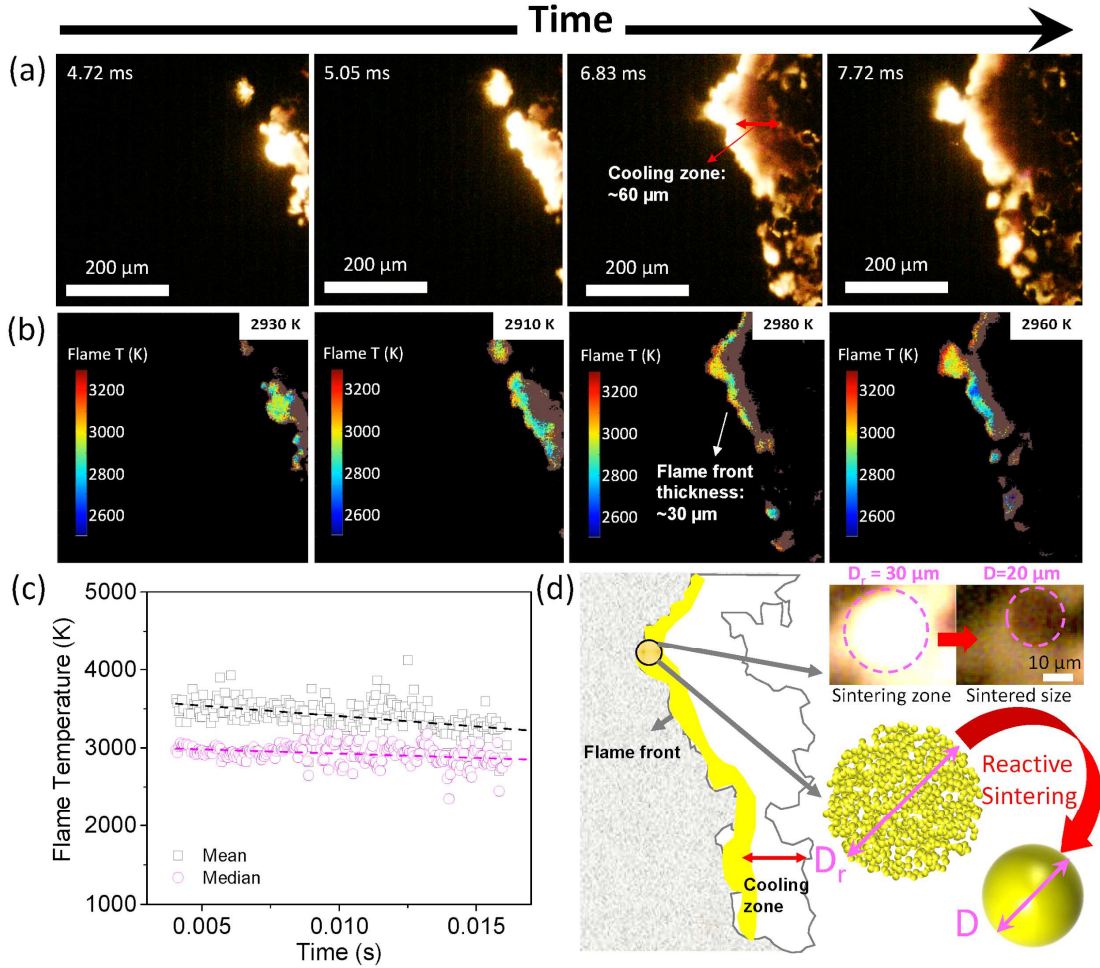
allowing exact identification of the area later in a scanning electron microscope (Figure 4-1e). A summary of the kind of information is shown in the panel of images above: Read from right to left are a snapshot taken from a single video frame (Figure 4-1b), the corresponding temperature map (Figure 4-1c) of the corresponding frame, temperature of single particle showing a thermal gradient within a particle (Figure 4-1d) and finally the same particle imaged under SEM with its EDS map (Figure 4-1e).

Figure 4-2a shows temporal reaction propagation snapshots of the Al/CuO nanothermites (90 wt.%) captured in a zone of  $512\ \mu\text{m} \times 512\ \mu\text{m}$  with a frame rate of  $\sim 18,000$  ( $55.5\ \mu\text{s}$  per frame) and the corresponding flame temperature maps were obtained by a color camera pyrometry through image processing. Four typical frames and their temperature maps were selected and shown in Figure 4-2a and Figure 4-2b, respectively, from which one can see the flame fronts consist of stochastic bright spots which discontinuously propagate the reaction. These bright areas are roughly divided into a leading flame front and a following cooling zone, as distinguished by the brightness. The noticeably brighter area spanning  $\sim 30\ \mu\text{m}$  was confirmed to be the leading edge from measured temperature  $\sim 3000\text{K}$  which is close to previously measured reaction temperature of Al/CuO.<sup>73</sup> As shown in Figure 4-2c, the time-resolved temperature was found to be constant at  $\sim 3000\ \text{K}$  (median) and  $\sim 3500\ \text{K}$  (mean) irrespective of flame front location, thus confirming that the reaction is relatively homogenous with a stable energy release despite the discontinuous front.

Reactive sintering, as discussed above and previously observed,<sup>47,58,152,153</sup> is the coalescence of aggregated and/or agglomerated nanoparticles driven by heat release during the reaction and results in the effective loss of nanostructure that was originally

introduced to enhance reactivity. As Figure 4-2d shows, after the flame front has passed any given area, sintered particles with a mean diameter of  $\sim 20\text{ }\mu\text{m}$  are observed, representing a three order-of-magnitude increase in particle size from the nano-sized precursors. It is also notable that the flame front thickness is roughly the same size as the sintering zone (as defined by a noticeably bright/hot area where reactive sintering is occurring), a reasonable observation considering that these flame fronts were constructed by networks of individually sintering particles propagating the reaction by either advection or heat release. As schematically shown in Figure 4-2d, the fact that the cooling zone is  $\sim 2\text{-}3$  times larger than the flame front thickness (=sintering zone) indicates the cooling time would be longer than the sintering time.

In Figure 4-2d, the area of the bright flame ball is the sintering zone which is effectively the flame front thickness. Considering the aggregate projected area (with a diameter of  $D_r$ ) is the sintering zone and the cooled particle ( $D$ ) is the final sintered size, the relationship between  $D_r$  and  $D$  can be explored. For these materials (see Figure 4-7 for SEM image) the density was measured to be 33% theoretical max density which is consistent with prior work.<sup>156–158</sup> At an aggregate packing density of 33%, coalescence will result in rough factor of 3 decrease in volume of the resulting sintered particle and 0.69 factor decrease in diameter (i.e.  $D \sim 0.69 D_r$ ). This result is remarkably consistent with the experimental results shown in the microscope image in Figure 4-2d which depict the final sintered size ( $\sim 20\text{ }\mu\text{m}$ ) and the sintering zone is  $\sim 30\text{ }\mu\text{m}$ . The sintering zone is thus, for a thermite, the effective thickness of the reaction zone and represents the local volume of material reacting as an entity, resulting in a single sintered particle product.



**Figure 4-2:** (a) Series of snapshots of burning flame front. Labelled on the left-top corner is the time after flame front entering the zone ( $512 \mu\text{m} \times 512 \mu\text{m}$ ). (b) The corresponding flame temperature maps and the average temperature (in Kelvin). High error points were marked as brown and are excluded from the calculations. (c) The mean (square marker) and median (round marker) flame temperature with time (with reference lines). (d) Schematic showing reactive sintering and its spatiotemporal relationship to reaction propagation. Note: the propagation direction is from right to left.

#### 4.3.2 Mapping optical to electron microscopy

To investigate the morphology, composition and size distribution of the post-combustion product, the reaction product-coated slide was examined by scanning electron microscopy (SEM) and energy dispersive spectrometry (EDS). As shown in Figure 4-3a and Figure 4-3b, an exact one-for-one correspondence can be found

between the in-operando microscope imaging and the SEM image. This is an important result because it enables post-reaction forensics for composition, corresponding to the exact reaction conditions observed in the in-operando microscope imaging. The average size of these spherical particles is  $\sim 25\ \mu\text{m}$ , confirmed by both optical and electron microscope. To see them more clearly, a typical particle was selected and marked as area A and B in Figure 4-3a and Figure 4-3b, respectively, which will be discussed in detail later.

Area A in Figure 4-3a was enlarged and the series of snapshots ( $55.5\ \mu\text{s}$  per frame) are shown in Figure 4-3c. At the time of this study, this series of images is believed to be the first observation of the temporal and spatial propagation of a thermite reaction at the microscale. Although the spatial resolution of this optical microscope is clearly inferior to an electron microscope, the image sequences are especially useful in that the whole reactive sintering process can be visually observed for the nanocomposite thermite, and from this, the sintering time, cooling time and sintered size can be obtained. If reactive sintering time is defined as that from ignition to the time of brightest emission, the entire sintering process takes  $\sim 170\ \mu\text{s}$  (3 frames) and the subsequent cooling time required an additional  $\sim 350\ \mu\text{s}$ . The bright part in the images is much larger than the final sintered particle, which confirms the above argument and consistent with the above calculation: that the reaction zone is significantly larger than flame front thickness. Additionally, from frame 2 to frame 3, the reaction front of the sintering moves  $\sim 30\ \mu\text{m}$  in  $\sim 55\ \mu\text{s}$ , corresponding to a propagation rate  $\sim 50\ \text{cm s}^{-1}$  for reactive sintering, a notably faster time than the macroscale flame velocity ( $\sim 3.3\ \text{cm s}^{-1}$ ) when tracking the flame front during one whole

burning event (Figure 4-8). This observation is further confirmed by the macroscale burn stick experiments (Figure 4-9), revealing a propagation relationship that is only evident when observed on the microscale (discussed below).

By zooming in, a single region that coalesces to a single particle (Figure 4-3c) can be imaged. Focusing on an instant in time when the emission is brightest (frame 3 in Figure 4-3c) enables measurement of both the size of the resulting sintered particle and its temperature;  $\sim 30\text{ }\mu\text{m}$  and  $\sim 2900\text{ K}$ , respectively. This is consistent with the reaction front thicknesses ( $\sim 30\text{ }\mu\text{m}$ ) discussed above and previously measured temperatures ( $\sim 3000\text{ K}$ ) in macroscale experiments. It is notable that a dynamic in-operando temperature measurement of a single sintering particle can be achieved on the resolution of  $\mu\text{m}$ . Furthermore, as Figure 4-3d shows, there is a  $\sim 1000\text{ K}$  temperature difference across the particle in a distance of  $\sim 30\text{ }\mu\text{m}$ , which indicating a temperature gradient of  $3 \pm 1 \times 10^7\text{ K m}^{-1}$ , and in the same direction as the flame propagation in this area.

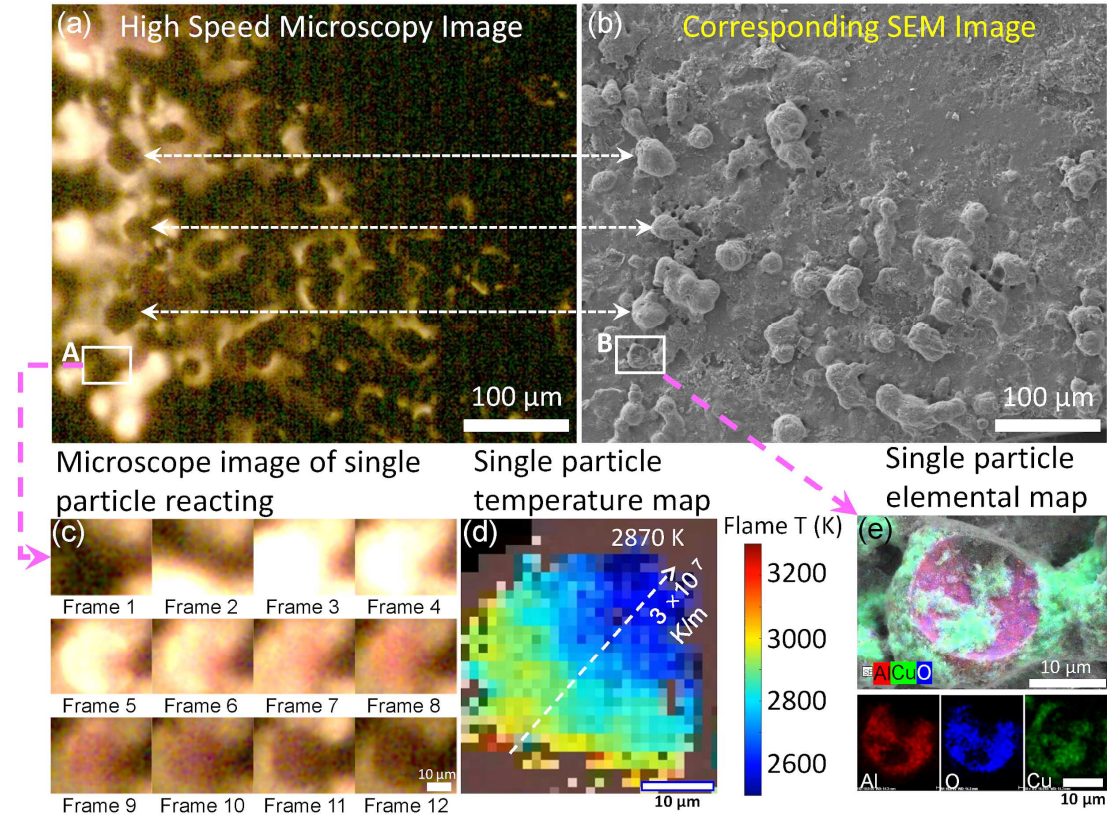
In area B of Figure 4-3b the corresponding particles after sintering were also enlarged and the high-resolution EDS maps are shown in Figure 4-3e. From them, it was concluded that the main composition of the sintered particle is  $\text{Al}_2\text{O}_3$  ( $\sim 20\text{ }\mu\text{m}$ ) and the coated smaller particles are Cu (in Figure 4-3e,  $< 20\text{ nm}$ ), which is confirmed by both point-mode EDS and X-ray diffraction patterns (Figure 4-10). Such small Cu nanoparticles indicates that Cu was vaporized during the reaction due to a flame temperature ( $\geq 2900\text{ K}$ ) above copper's boiling point ( $2835\text{ K}$ ). Another 10 particles in Figure 4-4b were also examined and the data was listed in Table 4-2. The average sintering time and cooling time of these 11 particles is  $\sim 170\text{ }\mu\text{s}$  and  $\sim 265\text{ }\mu\text{s}$ ,



respectively, consistent with the fact that cooling time is longer than sintering, resulting in a larger cooling zone than sintering zone as evident in Figure 4-2. We have tabulated the various length and time scales explored in this study in Table 4-1.

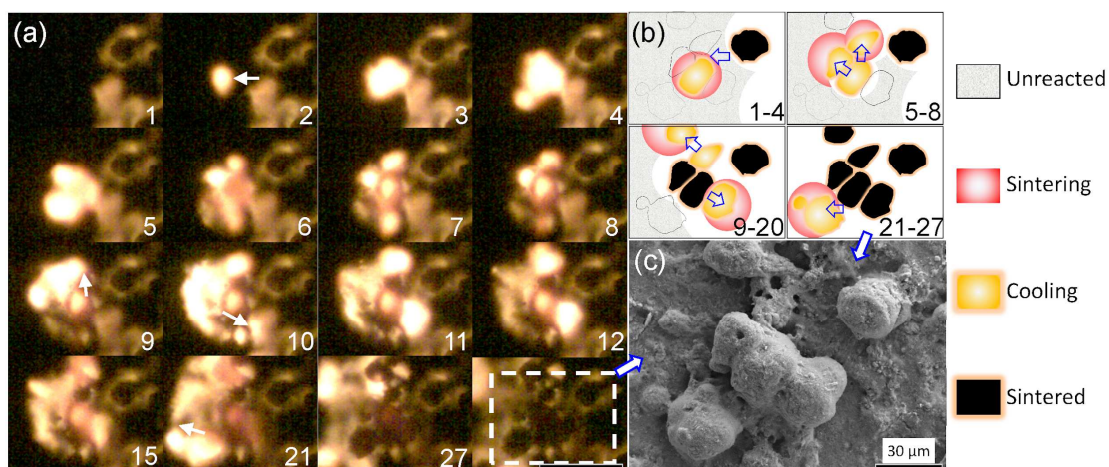
**Table 4-1:** Summary of different length and time scales in this study.

Length scale of flame propagation	Length scale of sintering	Time scale of sintering	Propagation rate
Flame front thickness: $\sim 30\ \mu\text{m}$	Sintering zone: $\sim 30\ \mu\text{m}$	Sintering time: $\sim 170\ \mu\text{s}$	Spatial propagation rate of reactive sintering: $\sim 50\ \text{cm s}^{-1}$
Cooling zone: $\sim 60\ \mu\text{m}$	Sintered size: $\sim 20\ \mu\text{m}$	Cooling time: $\sim 265\ \mu\text{s}$	Burning rate of the macro film: $\sim 3.3\ \text{cm s}^{-1}$



**Figure 4-3:** (a) Typical in-operando microscope image of Al/CuO reactive sintering, and after, (b) the corresponding SEM image of sintered combustion residue was found. (c) Series of snapshots of a single particle ( $55.55\ \mu\text{s}$  per frame) showing the appearance of a large particle indicative of reactive sintering. (d) Intra-particle temperature at peak temperature showing a thermal gradient within particle. (e) The EDS mapping of the above in-operando sintered  $\text{Al}_2\text{O}_3$  particle coated with Cu nanoparticles, as evident by EDS mapping (red: Al; blue: O; green: Cu). Note: combustion is from right to left.

As mentioned above and summarized in Table 4-1, the time scale of reactive sintering is as low as a few  $\mu\text{s}$  and the spatial propagation rate of reactive sintering is at least a factor of 10 faster ( $50$  vs.  $3 \text{ cm s}^{-1}$ ) than the observed macroscale flame velocity. To further explore this difference between micro and macro scales, a group of sintering particles were closely monitored as a bridge between reactive sintering and flame propagation. In Figure 4-4a, flame propagation is from right to left. An initial burning spot (frame 2) spreads to the surrounding area, where another two sintered particles appear. However, prior to cooling, the adjacent area on the left-top and right-bottom are observed to ignite and then move to the left. The schematic of the above process is demonstrated in Figure 4-4b and the final sintered particles are shown in Figure 4-4c. More image sequences and the EDS results of final particles can be found in Figure 4-11. Even though reactive sintering occurs in  $\sim\mu\text{s}$ , the propagation of the reaction is relatively slow and limiting by the heat conduction and high ignition temperature of Al/CuO ( $\sim 1000 \text{ K}$ )<sup>159</sup> which, in the microscope temporal image, show an almost stochastic behavior despite the overall reaction front moving in a given direction. This suggests that density gradients set up local sintering regions which comprise heat generation centers. These centers transport energy to neighboring areas that ignite. In this way propagation is very similar to the general concept of laminar flame theory in which flame velocity  $\sim (\text{thermal diffusivity} \times \text{reaction rate})^{1/2}$ . The difference being that the macro scale reaction velocity is limited by the thermal diffusivity between the sintered heat sources.

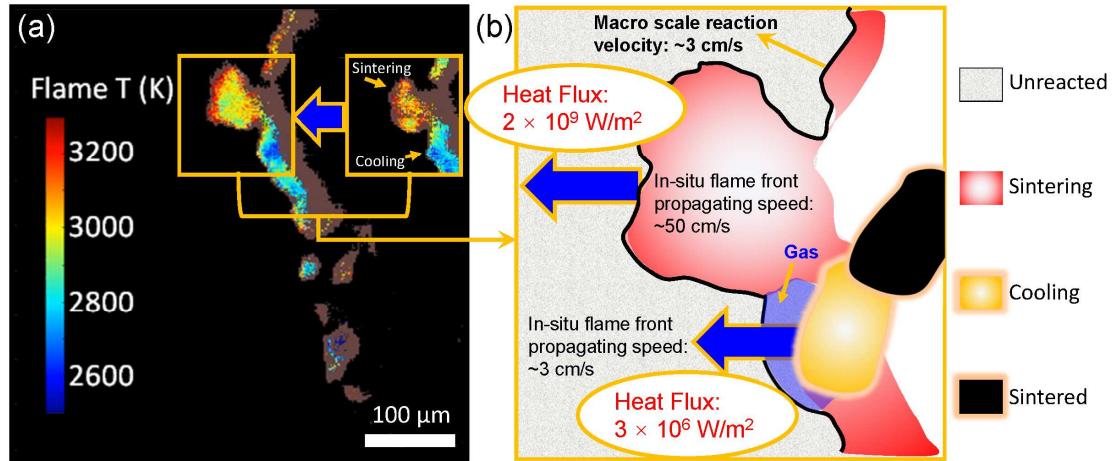


**Figure 4-4:** (a) Series of reactive sintering and ignition snapshots of a group of particles (labeled is frame sequence, 55.55  $\mu\text{s}$  per frame, scale bar: 100  $\mu\text{m}$ ) and (b) its corresponding schematic cartoon. Note: the marked arrows are propagating direction. (c) The corresponding SEM image of the sintered  $\text{Al}_2\text{O}_3$  particles coated with Cu nanoparticles, as evident by the EDS results in Figure 4-11.

#### 4.4 Discussion

To evaluate the above argument, consider that during the passing of the reaction front (1) reactive sintering, (2) cooling, and (3) a final sintered product is observed (see Figure 4-5). The fact that the flame front consists of these different stages imply an inhomogeneous reaction front with different heat fluxes. A heat flux can be estimated as  $\text{heat flux} \sim \text{thermal conductivity} \times \text{temperature gradient} = 10^9 \text{ W m}^{-2}$  in the reactive sintering stage based on the measured temperature gradient and an estimate thermal conductivity for  $\text{Al/CuO} \sim 60 \text{ W m}^{-1} \text{ K}^{-1}$ . This large high heat flux supports a rapid front propagating with a velocity as high as  $\sim 50 \text{ cm s}^{-1}$ . However, when reactive sintering has mostly completed (cooling stage), the heat flux declines by  $\sim 3$  orders of magnitude to  $10^6 \text{ W m}^{-2}$ , owing to much lower thermal conductivity of gas ( $0.1 \text{ W m}^{-1} \text{ K}^{-1}$ ) which separates the reacted from unreacted material. While local rapid reactions are seen, these reaction events are slowed down by low conductivity zones resulting in an overall propagation rate that is considerably lower. This calculation indicates that the heat flux

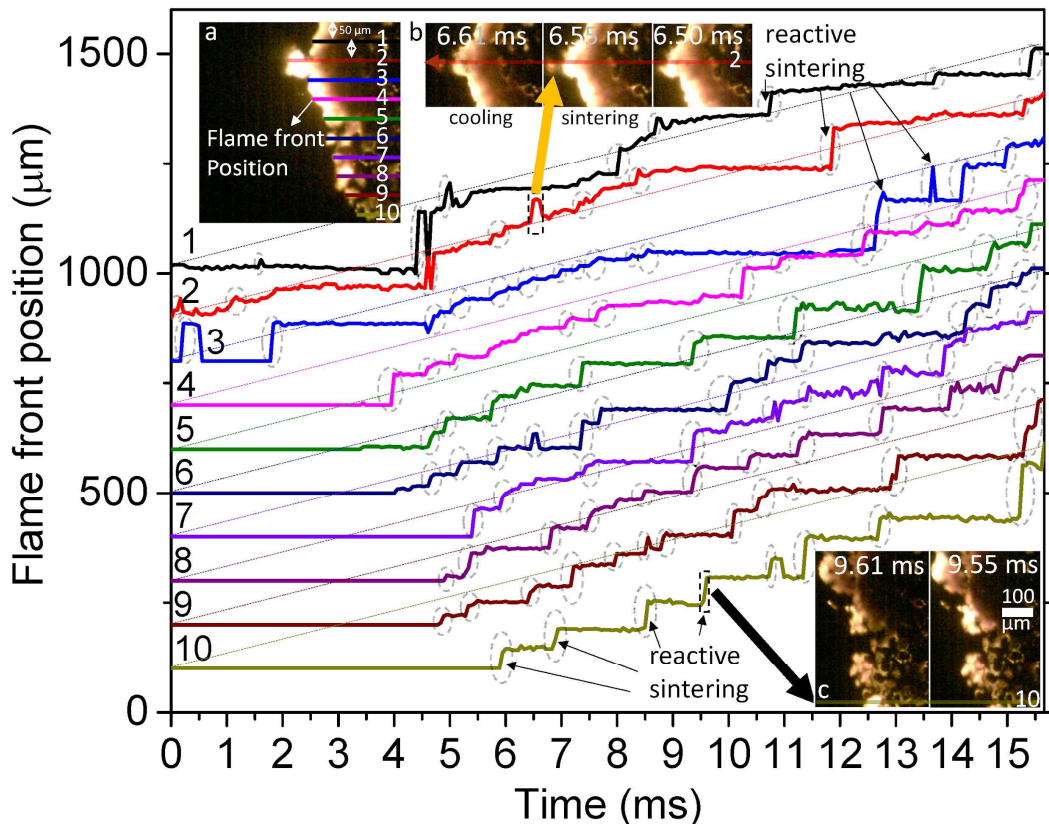
in the cooling stage in a reactive sintering process is critical to the macroscale flame velocity.



**Figure 4-5:** (a) Two typical temperature map snapshots of flame front of Al/CuO nanothermite and (b) its schematic showing of heat flux distribution from different stages of reactive sintering particles.

To confirm this, the time-resolved flame front positions were tracked every  $\sim 50 \mu\text{m}$  from top to bottom (10 positions in each frame) and the results are shown in Figure 4-6. From this, one can easily distinguish the reactive sintering stage, where the local flame front jumps in a near step function on the position-time curve. This is consistent with a previous conclusion that reactive sintering occurs much faster than the macroscale flame propagation. After each reactive sintering stage (Figure 4-6b and Figure 4-6c), a cooling stage (where the flame front position slightly retracts, see insert Figure 4-6b) always follows. The different heat fluxes by these two stages drives the flame front propagation at different velocities and, as proposed above, results in a discontinuous flame front. Eventually, the macroscale flame velocity is limited by the cooling stage, which is evident in Figure 4-6 since all of the curves have roughly similar slopes.





**Figure 4-6:** Tracking the time-resolved flame front position (propagation direction is from right to left). Left-top insert (a) shows, location where flame front is tracked and plotted and ordered 1-10, separated by 50  $\mu\text{m}$ . Insert b and c show two typical “reactive sintering” events. The “reactive sintering” are annotated with the dashed circles on the line plots.

In this paper, high-speed microscopy/thermometry enables observation of the in-operando micro reaction of Al/CuO nanocomposite thermites. The proposed mechanism of reactive sintering in nanothermites could be visualized and confirmed in high spatial ( $\sim 1 \mu\text{m}$ ) and temporal ( $\sim 55 \mu\text{s}$  per frame) resolutions. The temperature map of the flame fronts and the sintering particles were also obtained by a color camera pyrometry of the same images. The flame front thickness was determined from the above results as  $\sim 30 \mu\text{m}$  and the average reactive sintering time and sintered size were obtained as  $\sim 170 \mu\text{s}$  and  $\sim 25 \mu\text{m}$ . Post reaction analysis of the exact same area and the corresponding sintered particles were found in scanning electron microscope (SEM)

which enables one-to-one composition mapping. For these studies, it can be concluded that the local reaction velocity is an order of magnitude higher than macroscale flame velocity. These results imply that local heat generation to create sintered particles are limited in their heat transport to neighboring areas and is likely the rate limiting step. Heat flux calculations indicate that the macroscale flame velocity was highly dependent on the heat release rate in the cooling stage in a reactive sintering process.

## **4.5 Methods**

### **4.5.1 Chemicals and precursors**

METHOCEL™ F4M Hydroxypropyl Methylcellulose (HPMC) was obtained from Dow Chemical Company. Polyvinylidene fluoride (PVDF, average molecular weight: ~534,000) and N, N-Dimethylformamide (DMF, 99.8%) were purchased from Sigma-Aldrich. CuO nanoparticles (~40 nm) were purchased from US Research Nanomaterials. Aluminum nanoparticles (Al NPs, ~85 nm) were purchased from Novacentrix. The active aluminum content is ~81 wt. % according to thermogravimetry analysis (TGA). All the chemicals were used as received. When preparing a precursor, 125 mg HPMC and 125 mg PVDF were first dissolved in 8 mL DMF and magnetically stirred for ~2 hrs to get a clear solution. Then 1761 mg CuO and 489 mg Al NPs were dispersed into the above polymer solution by ultrasonication for ~1 hr. The prepared slurry was then magnetically and mechanically stirred for 24 hrs and 1.5 hrs, respectively, and the precursor was ready to print.

#### 4.5.2 3D-printing on glass slides

The as-prepared precursor was printed using a Hyrel 30M 3D printer to a commercial cover glass slide (VWR Company, 0.17mm thickness, 22mm square) pre-heated to 80°C. The printing path was an 8 cm  $\times$  8 cm domain at  $\sim 0.3$  mL min<sup>-1</sup> and writing speed of  $\sim 22$  cm min<sup>-1</sup>.

#### 4.5.3 Additional information about high-speed microscope imaging system

Figure 4-1a shows the configuration of the high-speed microscope with the high-speed video camera and a 40x microscope objective (Nikon) with a working distance of 0.66mm and a high numerical aperture of 0.75. The light collimated by the microscope passes through a beam splitter (ThorLabs) and is focused by the camera lens (Nikon 105mm Macro) focused at infinity. The third port of the beamsplitter cube houses a red LED (630 nm), which is collimated using a plano convex lens at  $1f$ . The collimated beam is reflected by the beam splitter and focused on the sample via the microscope and the scattered light from the sample is imaged by the camera for focusing purposes. The high-speed video camera (Vision Research VEO710) recorded at framerates of 18,000 frames s<sup>-1</sup> with an exposure of  $\sim 55$   $\mu$ s.

#### 4.5.4 Color video pyrometry

Color ratio pyrometry was performed to estimate temperature of the burning films.<sup>73</sup> By taking ratios of raw color channel intensities, as imaged by a red/green/blue Bayer filter and CMOS assembly, dependency on most variables associated with light intensity (e.g. solid body angle) are eliminated except for those pertaining to channel gain, emissivity, and spectral response of the optical components (microscope lens,

camera lens, beam splitter, Bayer filter, CMOS sensor) at individual wavelengths. The camera was calibrated for temperatures ranging from 773-4773 K with a Newport Oriel 67000 Series Blackbody Infrared Light Source using Planck's Law and the associated graybody assumption. MATLAB was used to extract raw pixel values, demosaic the Bayer filter to recover values for RGB at each pixel, and calculate temperatures based on color intensity ratios of the pixels. Three color ratios (green/red, blue/green, and blue/red) were simultaneously used to estimate temperature by minimizing their summed error from theoretical ratios with a nominal error less than  $\sim 110$  K. For the figures that show temperature of a single sample as a function of time, only unsaturated pixels above the black level and within the error threshold are used to report mean/median temperature of the frame for a contiguous area of at least 10 acceptable pixels.

#### *4.5.5 Morphology characterization*

The microstructure of the printed samples was investigated using a Hitachi SU-70 scanning electron microscope (SEM) coupled to an energy dispersive spectrometer (EDS). The printed samples were sectioned in liquid nitrogen and attached to a carbon film on an SEM stage. The SEM and EDS mapping images of 90 wt.% Al/CuO nanothermites were shown in Figure 4-7.

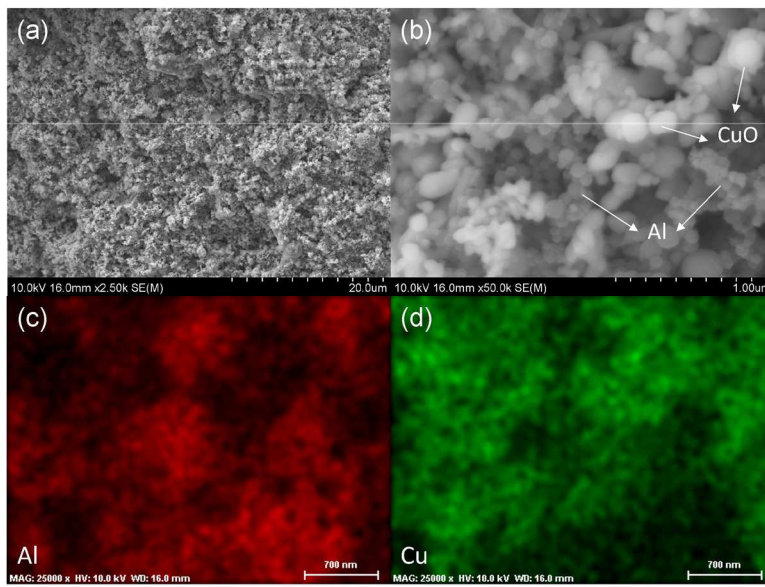
#### *4.6 Acknowledgements*

This work was supported by the AFOSR. We acknowledge the support of the Maryland Nanocenter and its NispLab. The NispLab is supported in part by the NSF as an MRSEC Shared Experimental Facility. The authors wish to thank Dr. Rohit Jacob

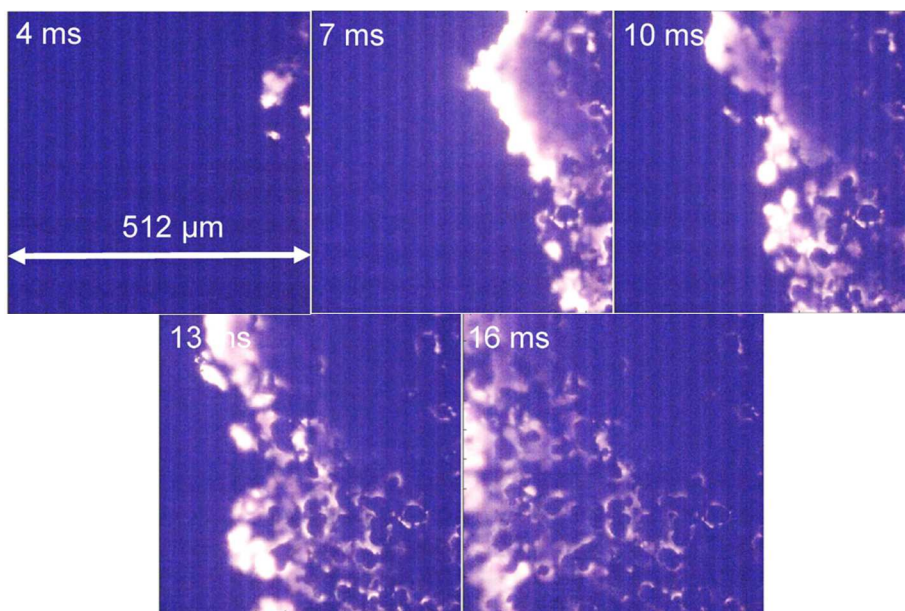


for setting up the initial version of this experiment while a graduate student in the group. Reproduced from H. Wang\*, D.J. Kline\*, M.R. Zachariah, In-operando high-speed microscopy and thermometry of reaction propagation and sintering in a nanocomposite, Nat. Commun. 10 (2019) 3032. doi:10.1038/s41467-019-10843-4. with permission from Springer Nature. \*Denotes that authors contributed equally to this work.

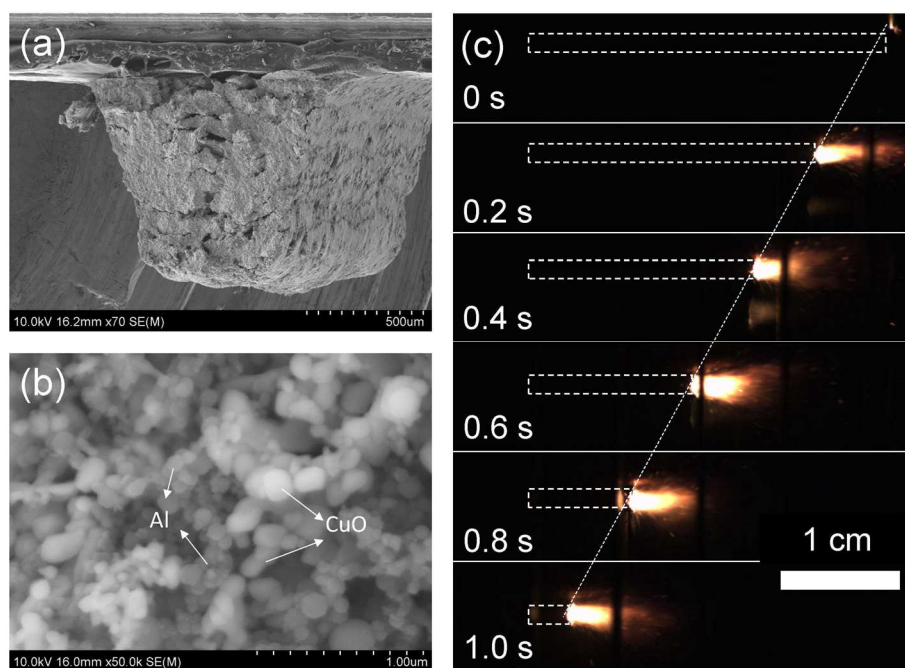
#### 4.7 Supplementary Information



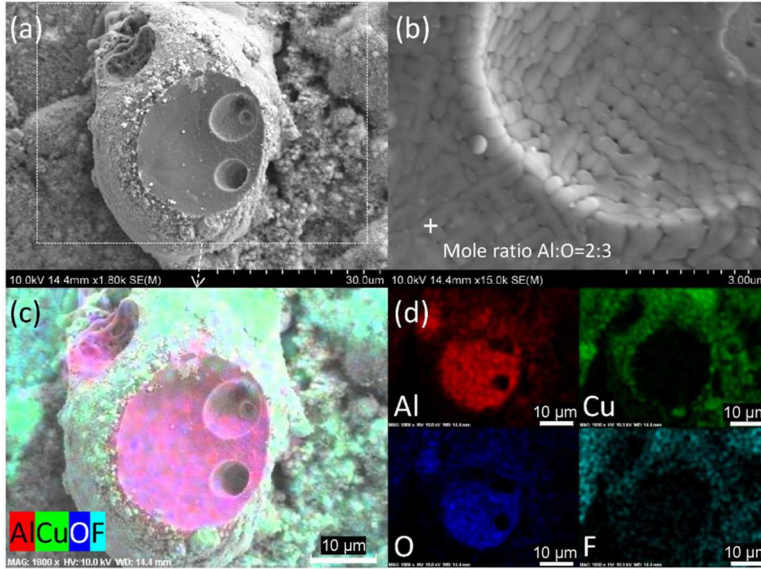
**Figure 4-7:** Low (a) and high (b) SEM images, EDS results (c, Al; d, Cu) of a cross-sectional Al/CuO/HPMC/PVDF film.



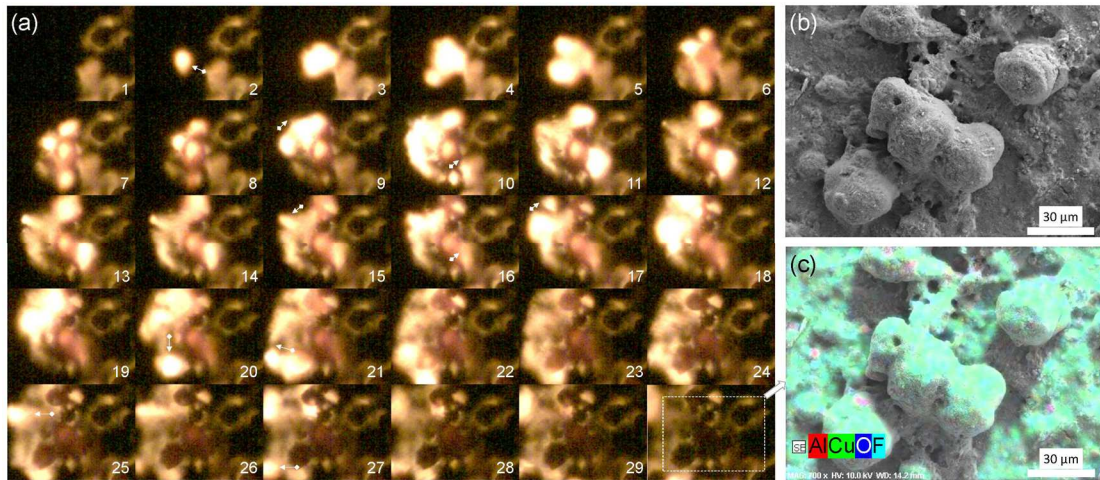
**Figure 4-8:** The flame propagation observed in a  $512\ \mu\text{m} \times 512\ \mu\text{m}$  zone.



**Figure 4-9:** Low (a) and high (b) resolution SEM images of the printed Al/CuO (90 wt.%) stick, and its burning snapshots in argon atmosphere (1 atm).



**Figure 4-10:** Low (a) and high (b) SEM images, EDS results (c and d) of a cross-sectional sintered particle. Based on a simplified estimation, one ~25-micron sintered  $\text{Al}_2\text{O}_3$  was formed by ~30 million ~80 nm Al NPs. The fine  $\text{Al}_2\text{O}_3$  inside (~300 nm) was sintered by ~50 Al NPs.



**Figure 4-11:** Series of reactive sintering and ignition snapshots of a group particles (a, labeled is frame sequence, 55.55  $\mu\text{s}$  per frame). Note: the marked arrows are propagating direction. The corresponding SEM image (b) of the sintered  $\text{Al}_2\text{O}_3$  particles coated with Cu nanoparticles, as evident by the EDS results (c).

**Table 4-2:** Sintering time, cooling time and size distribution of typical particles. Note: The cooling time for particle 1#, 2# and 4# is invalid because they were reheated by surrounding sintering particles.

Particles #	Sintering time (us)	Coolingtime(us)	Size (um)
1	170	1780 (invalid)	35
2	170	1220 (invalid)	20
3	110	110	35

4	220	1060 (invalid)	30
5	220	280	20
6	170	390	25
7	110	170	15
8	220	440	35
9	170	170	20
10	220	280	25
11	110	280	20
Mean	170	265	25
Error	50	110	7

## Chapter 5: Experimental observation of the heat transfer mechanisms that drive propagation in 3D printed reactive materials

### *5.1 Summary*

The rise of additive manufacturing has led to the development of new architectures and allowed researchers to 3D-print a variety of energetic composites with different formulations. However, reactive sintering within the composites can negatively impact overall performance of nanoenergetic or reactive materials via the loss of nanostructure. It has been previously proposed that the addition of a gas generator could reduce the impact of reactive sintering while enhancing reactivity. Iodine pentoxide ( $\text{I}_2\text{O}_5$ ) is an extremely powerful oxidizer and a strong gas generator which has been proposed as an efficient source of molecular iodine for biocidal applications. However, fabricated  $\text{Al}/\text{I}_2\text{O}_5$  nanocomposite propellants are unable to propagate. In this study, it is demonstrated that the addition of small quantities of  $\text{CuO}$  enables consistent propagation. Employing high-speed pyrometry and microscopy, advective heat transfer in the system is estimated and it is demonstrated that the addition of  $\text{CuO}$  offers a pathway for enhanced energy-transfer by metal condensation that is not available when  $\text{I}_2\text{O}_5$  is the sole oxidizer. Elimination of condensing metal vapor as a participant in energy transfer increased the reliance of the system on particle advection. Small additions of  $\text{CuO}$  yield a condensable product vapor which can effectively eliminate the reliance on heat transfer via advection, enabling consistent propagation in reactive solid materials. A general conclusion is made that where

reliable propagation is critical, such as a stable burn of a solid propellant or delivery of a biocide, small additions of an oxidizer with a condensable metal vapor could be the difference between guaranteed performance and critical failure.

## **5.2 Introduction**

Thermite (or reactive materials) are a subclass of energetic materials that participate in a highly exothermic redox reaction and have been traditionally prepared using a mixture of metals and metal oxides.<sup>6,24</sup> Compared to organic, monomolecular explosives, thermite reactants have a higher energy density because the enthalpy of formation for solid products is substantially lower than the gaseous products of organic explosives (whose primary products are CO<sub>2</sub> and water).<sup>6,21,22,24</sup> However, thermites have been traditionally mixed using micron powders which have large diffusion distances between the fuels and oxidizers and thus have much slower reactions than explosives.<sup>38</sup> Recent research in thermites has centered around the use of nanoparticles as the fuel and oxidizer to increase the burn rate<sup>6,27,38,55,160</sup> and the rise of additive manufacturing has opened up new avenues by which to architecture<sup>35,78,86,87,90</sup> these materials for applications in high energy-density solid propellants.

Nanothermites have also been investigated as a biocide delivery mechanism for defense against the threat of bioweapons.<sup>74,161,162</sup> Iodine-containing compounds have received the widest attention as a readily-available biocidal additive to reactive materials due to the effectiveness of molecular iodine at killing spores.<sup>74,91,161–164</sup> Numerous studies have focused on the incorporation of iodine pentoxide (I<sub>2</sub>O<sub>5</sub>) into a variety of nanothermites as both the oxidizer and iodine carrier where it has been found that the combustion performance in a constant-volume cell is superior to metal/metal



oxide mixtures, likely due to the large amount of gas generated as part of the reaction.<sup>54,74,165</sup> Other studies have investigated the incorporation of iodine into the oxide shell of the metal fuel particles to further enhance reactant mixing and improve performance.<sup>91,163</sup>

Gas-producing oxidizers, such as  $\text{I}_2\text{O}_5$  and  $\text{CuO}$ , also offer a unique advantage over the strictly condensed-phase oxidizers since they reduce reactive sintering during the thermite reaction – a phenomenon that results in the loss of nanostructure during reaction and can arrest reactions prematurely.<sup>47,58,73,166</sup> By introducing a gas generator that also serves as an oxidizer for the mixture, agglomeration of particles is reduced. Furthermore, the gas release would also likely increase the heat transfer via advection to unreacted areas of the propellant – a critical mechanism of heat transfer for sustained propagation in energetic materials as described by Egan et al.<sup>11</sup> However, Egan et al. also demonstrated that condensing metal vapor could contribute more than enough energy to sustain combustion, but it is estimated that only ~15% of the energy used for ignition is from product condensation and ~50% is from advective heat transfer in a modeled burn tube experiment.<sup>11</sup> Certainly, advection would be enhanced in the case of  $\text{I}_2\text{O}_5$  oxidizer over other metal oxides because of its considerably higher gas production, though it is unclear whether the increase in gas production could offset the energy lost in systems that totally lack the recondensing metal vapor.

This work investigates the role of different heat transfer mechanisms that facilitate propagation in solid propellants to improve combustion performance without compromising reliability. Propagation of the reaction front is observed at the microscale for high-loading, additively manufactured energetic materials to estimate

the impact of gas release, advection, and metal vapor condensation on combustion performance and stability. It was found that, while gas generation in the Al/I<sub>2</sub>O<sub>5</sub> reaction reduced reactive sintering, the material was unable to consistently propagate, likely attributable to the lack of condensing metal vapor products that would provide heat feedback for propagation. However, small additions of CuO to the sample improved the propagation reliability without significant impacts on the reactive sintering, thus revealing the fine balance between gas generation and heat transfer as tuning mechanisms for performance in solid energetic materials.

### **5.3 Methods**

#### **5.3.1 Chemicals**

METHOCEL F4M hydroxypropyl methylcellulose (HPMC) was ordered from Dow Chemical Company. Poly(vinylidene fluoride) (PVDF, molecular weight:  $\approx 534,000$ ), N,N dimethylformamide (DMF, 99.8%) and Iodine Pentoxide (I<sub>2</sub>O<sub>5</sub>) were purchased from Sigma-Aldrich. CuO nanoparticles ( $d \approx 40$  nm) were purchased from US Research Nanomaterials. Aluminum nanoparticles (Al NPs,  $\approx 85$  nm) were purchased from Novacentrix. The active aluminum content is  $\approx 81$  wt% according to thermogravimetry/differential scanning calorimetry (TGA/DSC) results. I<sub>2</sub>O<sub>5</sub> crystals were purchased from Sigma Aldrich (99.9% purity). The crystals were heated in a tube furnace at 300°C for 2 hours, then ball-milled (Retsch Cryomill) with 5/32" steel balls in hexane for 2 hours to prevent moisture uptake. The powder was stored in a low-humidity environment and reheated in a tube furnace at 300°C for 1 hour immediately prior to ink preparation.



### 5.3.2 Precursor preparation

Inks for 3D-printing of burn sticks were prepared by first mixing 50mg PVDF and 75mg HPMC in 4mL of DMF for 30 minutes at 2000 rpm in a planetary centrifugal mixer (Thinky USA). The Al NPs were then added to the polymer solution and dispersed by ultrasonication for 35 minutes. After the dispersion of the Al NPs, the mixture was mixed in the Thinky Mixer again at 2000 rpm on the mixing mode for 5 minutes and placed in the sonicator for an additional 35 minutes. This process of addition, dispersion by ultrasonication, and mixing was repeated for the CuO NPs and I<sub>2</sub>O<sub>5</sub>.

Mixture formulations for the thermites added to the PVDF/HPMC mixture were calculated for the stoichiometric reaction between active fuel and solid oxidizer, whereby the oxidizer content was varied by small additions of CuO. The amount of CuO added was determined by assuming that a percentage of the active fuel was reacting with different oxidizers. Table 5-1 shows the formulations for the different precursors where the Al wt.% includes the mass of the Al<sub>2</sub>O<sub>3</sub> shell. Experiments were only performed on mixtures containing up to 25% CuO to retain the primarily biocidal properties of the I<sub>2</sub>O<sub>5</sub> mixture.

Formulations are named by the theoretical percentage of Al that would react with the different oxidizers. For example, in the Al/75% I<sub>2</sub>O<sub>5</sub>/25% CuO sample, 75% of the available aluminum would stoichiometrically react with the I<sub>2</sub>O<sub>5</sub> and the remaining 25% of Al would react with CuO. Mixture formulations did not consider the stoichiometry of the samples including the fluoropolymer binder, therefore the overall sample stoichiometry is slightly fuel-lean ( $\Phi \sim 0.92-0.93$ ).

**Table 5-1:** Mixture formulations of thermites added to PVDF/HPMC ink. Al wt.% includes the mass of Al<sub>2</sub>O<sub>3</sub> shell.

Oxidizer (% Al reacting with I <sub>2</sub> O <sub>5</sub> )	Al (wt. %)	I <sub>2</sub> O <sub>5</sub> (wt. %)	CuO (wt. %)
100%	25%	75%	0%
95%	25%	71%	4%
75%	25%	54%	22%

### 5.3.3 *Direct-writing of burn sticks*

Burn sticks were fabricated using a Hyrel System 30M printer with a heating stage. The precursor was loaded in a 30 mL syringe with a needle size of 18 gauge (inner diameter 1 mm). The heating stage was topped with a glass, microscope coverslips (VWR 0.17mm thickness, 22mm square) and heated to  $\approx 70^{\circ}\text{C}$  before printing to ensure total solvent evaporation before deposition of another layer. The width of the deposited sticks was kept at 2 mm by employing an 18 gauge needle.<sup>35</sup> The printer was programmed to write 14 layers, resulting in a thickness range of 388-450  $\mu\text{m}$  per burn stick. The thickness and cross-sectional area of the final products were measured by scanning electron microscopy (SEM). Density of the final products was estimated by using the measured mass and volume of the burn sticks. Porosity was calculated by using the measured and theoretical density values of the composites. The results are tabulated in Table 5-4.

### 5.3.4 *Macro-scale burn tests*

To evaluate the combustion performance of the prepared materials on the macro-scale, as-printed samples were removed from the surface of the microscope coverslips to propagate as a free-standing propellant. The burn performance apparatus is a glass cylinder with rubber stops at each end for atmosphere control and electrical

throughputs for combustion initiation. Evenly-spaced metal plates around the glass tube are used for pixel/distance ratio calculations for flame speed analysis. The free-standing films are mounted on one end between a braided nichrome wire and double-sided tape on the other end. The wire is resistively heated to ignite the sample, and the combustion event recorded using a high-speed camera (Phantom Miro M110) recording at 4,000 frames/s. Burn rate is calculated by tracking the distance that the reaction front moves as a function of time. The burn rate for samples that did not consistently propagate was measured for the period of time that a stable burn was sustained. All experiments were performed in a constant-pressure air atmosphere.

#### 5.3.5 High-speed in-operando microscopy/thermometry and particle tracking

Sample combustion was also observed using a custom-built high-speed microscopy apparatus that allows for reaction front imaging at  $\mu\text{m}$  and  $\mu\text{s}$  resolution.<sup>166</sup> The glass coverslips on which the samples are printed are mounted on a 3-axis translational stage, brought into focus, and ignited using a resistively heated nichrome wire along one edge. Propagation of the reaction front is then captured using a high-speed camera (Phantom VEO710L) that is coupled to a long-distance microscope (Infinity Photo-Optical Model K2 DistaMax). The apparatus provides a pixel/distance ratio of  $\sim 1.7 \mu\text{m}/\text{pixel}$  from a working distance of  $\sim 54 \text{ mm}$ . Videos of the reaction propagation were recorded at 14,000-18,000 frames/s.

The high-speed microscope apparatus has also been calibrated for color pyrometry to estimate the temperature of the reaction. Color pyrometry has been discussed in other articles and is only briefly described here.<sup>73</sup> Three channel intensity ratios are used to estimate the temperature of materials based upon the theoretical

intensity for the sensor assuming graybody emission behavior of the material. Calibration factors for the assembly are determined for the assembly by comparing the theoretical response (estimated using Planck's Law and the response spectrum of the CMOS sensor) to actual response when observing a blackbody source (Mikron M390). The calibration curve for this assembly can be seen in Figure 5-5. Prior to temperature calculations, the raw images are imported using a house-built MATLAB routine and demosaiced for the camera's Bayer filter using the built-in MATLAB algorithms.<sup>167</sup> For these videos, the error threshold was increased to a maximum error of 400K, however, the error is nominally 200K.

Particle tracking and estimated particle velocities of advecting materials produced by the reaction were estimated using the pixel/distance ratios and exposure times of the images for each video.

#### *5.3.6 Morphology characterization*

Scanning electron microscopy (SEM) was performed on as-prepared samples and post-combustion products to evaluate the morphology. A FEI NNS450 SEM and an energy-dispersive X-ray spectroscopy (EDX) probe were used to observe the sectioned samples and construct elemental maps of samples that were mounted with carbon tape to SEM stages.

### **5.4 Results**

#### *5.4.1 Materials*

Prior to combustion experiments, it was important to verify that the materials being prepared retained the properties of the original reactants – especially  $\text{I}_2\text{O}_5$ .  $\text{I}_2\text{O}_5$

has been known to absorb water, in which case its crystal structure changes to that of  $\text{HI}_3\text{O}_8$ . The as-purchased  $\text{I}_2\text{O}_5$  crystals were substantially larger than the nanometal additives and contained small amounts of  $\text{HI}_3\text{O}_8$ .<sup>168</sup> Therefore, crystals were dried in a furnace and ball-milled in hexane to prevent moisture uptake. This reduced the final size of the particles to  $\sim 2\text{-}4\ \mu\text{m}$  crystals, as shown in Figure 5-6, and prevented moisture uptake.<sup>168</sup>

An example image of the as-prepared films can be seen in Figure 5-6 for the original  $\text{I}_2\text{O}_5$  crystals,  $\text{Al/I}_2\text{O}_5$ , and  $\text{Al/75\% I}_2\text{O}_5\text{/25\% CuO}$  burn sticks. The images show that the  $\text{I}_2\text{O}_5$  crystals were reduced in size from  $\sim 25\text{-}100\ \mu\text{m}$  to  $\sim 2\text{-}10\ \mu\text{m}$  crystal domains that can be seen extending from the surface of the material. Despite the size reduction for the  $\text{I}_2\text{O}_5$  crystals, the Al and CuO primary particles are still substantially smaller ( $\sim 50\text{-}80\ \text{nm}$ ) in comparison. The final films are also porous primarily due to the evaporation of the DMF solvent used for the ink during the printing process. To further reduce the likelihood that the samples reabsorbed any water and eliminate any remaining solution, the samples were placed on a hot plate at  $100^\circ\text{C}$  prior to preparation and were stored in a desiccator if not used immediately. The sample moisture content up until the time of the experiments was considered minimal but was not directly measured.<sup>169</sup>

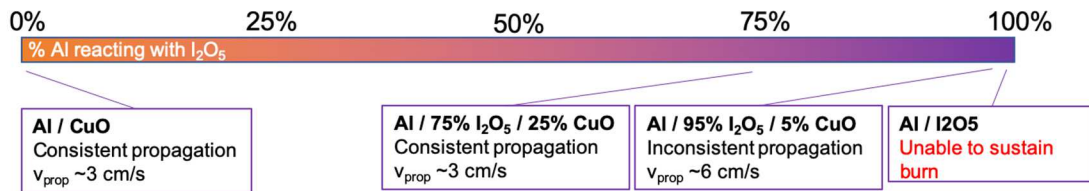
#### 5.4.2 Macro-scale burn tests

Free-standing  $\text{Al/I}_2\text{O}_5$  high-loading burn sticks would consistently ignite but would fail to propagate to completion – a peculiar result for a mixture that nominally should have excellent performance because of its enhanced gas generation, high adiabatic flame temperature, and low ignition temperature. Egan et al. detailed the role

of different heat transfer mechanisms on propagation in reactive materials that were packed into constant volume burn tubes.<sup>11</sup> The major result was that, while convection played a significant role on heat transfer for propagation, the presence of condensed metal vapor in the products was a comparable contributor. In the case detailed here (Al/I<sub>2</sub>O<sub>5</sub>), heat transfer via condensation is not as significant since Al<sub>2</sub>O<sub>3</sub> is not formed from the vapor and I<sub>2</sub> has a small heat of condensation.<sup>11</sup> A more detailed discussion on this result and the role of different heat transfer mechanisms on propagation in these materials is presented in a later section.

If the conjecture above is true, it should be possible to overcome the difficulties in propagation with the addition of material that could produce a condensable metal vapor. In this study, CuO was chosen to be added in small amounts such that a certain percentage of the available Al would stoichiometrically react with I<sub>2</sub>O<sub>5</sub> and the remainder would stoichiometrically react with CuO. Upon addition of 5% CuO (Al/95% I<sub>2</sub>O<sub>5</sub>/5% CuO), the sample propagated to completion in ~50% of all experiments with a propagation velocity of ~6 cm/s. Addition of more CuO (up to 25%) results in consistent propagation, albeit with a slightly lower velocity of ~3 cm/s. The propagation velocity of Al/75% I<sub>2</sub>O<sub>5</sub>/25% CuO is more reflective of the high-loading Al/CuO films that were tested by Wang et al.<sup>35,166</sup> A summary of the results is presented in Figure 5-1. Temperature-jump, time-of-flight mass spectrometry (T-Jump/TOFMS) was performed on oxidizer mixtures and revealed no apparent interactions between CuO and I<sub>2</sub>O<sub>5</sub> that would promote decomposition or gas release (see Table 5-2). Furthermore, adiabatic flame temperature calculations (with CHEETAH) shown in Figure 5-7 demonstrates that the adiabatic flame temperature at

constant pressure depresses slightly with the addition of CuO, implying that small additions of CuO do not significantly impact the overall thermophysics.<sup>9</sup> It is believed that, should the Al/I<sub>2</sub>O<sub>5</sub> sample been capable of sustaining combustion, it would likely have propagated at the fastest rate since Al/I<sub>2</sub>O<sub>5</sub> powders are known to burn faster than Al/CuO.<sup>75,116</sup> The apparent decrease in propagation velocity is therefore not a result of special interactions of the mixture, but rather reflects a sliding scale in performance from the pure Al/I<sub>2</sub>O<sub>5</sub> to the Al/CuO case.



**Figure 5-1:** Summary of combustion performance data from macroscale combustion experiments for different mixtures of Al/I<sub>2</sub>O<sub>5</sub>/CuO.

Whereas it would regularly be expected that the large volume of gas generated by I<sub>2</sub>O<sub>5</sub> during reaction would dramatically reduce reactive sintering leading to superior performance, an energy deficiency emerges which leads to poor or non-propagation. The enhanced burn with small additions of CuO and the ability to make propagation consistent suggests that condensing metal vapor may play a significant role in combustion of these samples and will be explored later in this paper.

#### 5.4.3 High-speed in-operando microscopy

To further investigate energy transfer mechanisms that drive propagation in reactive materials, it was important to see how the material was reacting on the particle scale. Egan et al. suggested that advection of hot material from the reaction front to regions far away must provide significant energy contributions to sustain propagation in energetic materials to make up for an energy deficit that was present in their

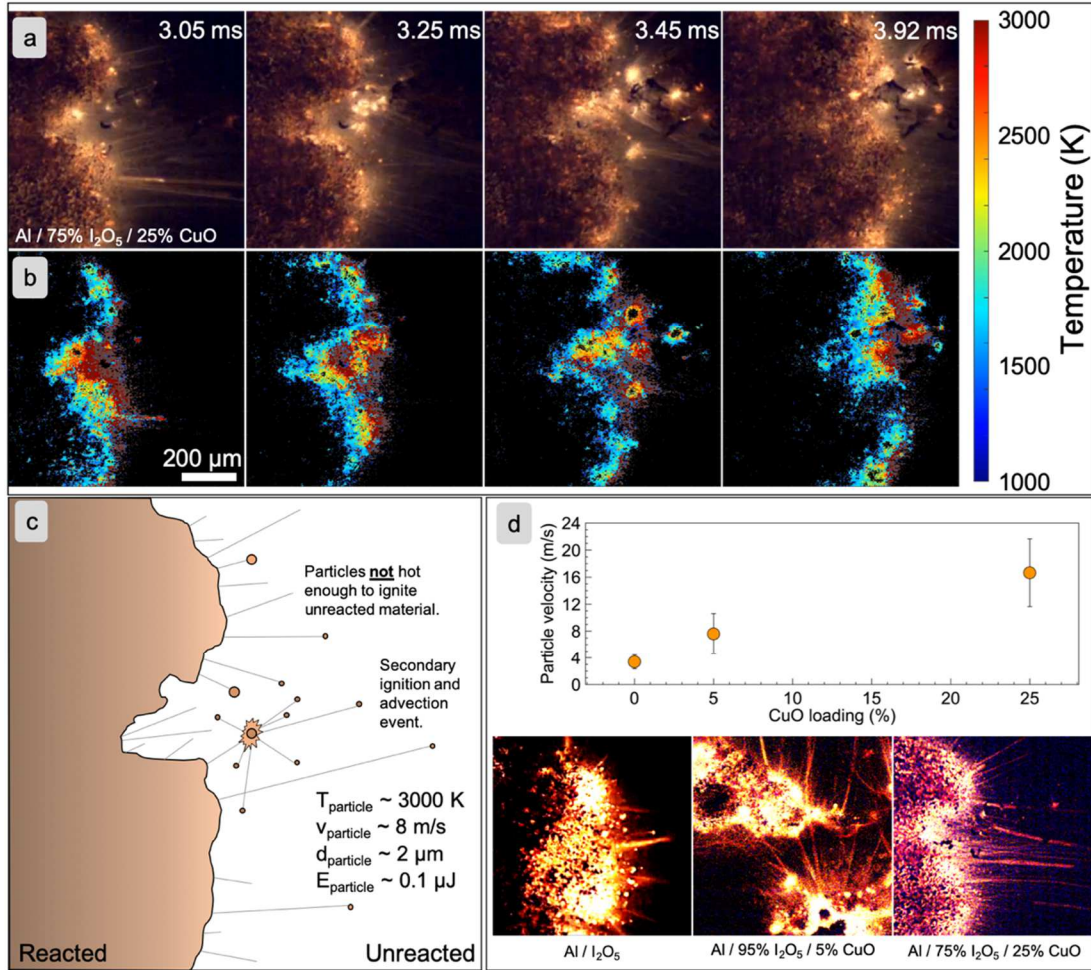
calculations.<sup>11</sup> When reaction propagation was visualized in our recent article, there were no significant advection events for a high-loading Al/CuO ink, however there was substantially less gas generated from that composite in comparison to Al/I<sub>2</sub>O<sub>5</sub> composites.

High-speed microscopy images of all of the Al/I<sub>2</sub>O<sub>5</sub>/CuO energetic composites tested showed clear particle advection events that were in stark contrast to the images that were presented in a previous study.<sup>166</sup> Each of the images collected in all of the mixtures had large concentrations of hot particle streaks that would either extinguish (if the final destination did not have a high concentration of advected particles) or would create a secondary ignition event at a location away from the front (depicted in Figure 5-2). The reaction fronts observed in these mixtures are also very different morphologically from those presented in previous work. The reaction front in these mixtures is not clearly-defined, likely due to the large gas generation that could distort the morphology at the film near the front.<sup>166</sup>

Reaction temperatures for the propagation event were also processed using color camera pyrometry<sup>73</sup> to estimate the bulk reaction temperature and advected particle temperature. Figure 5-2a-b show a typical reaction propagation event for the Al/75% I<sub>2</sub>O<sub>5</sub>/25% CuO mixture and its corresponding temperature map, which shows a wide temperature distribution ranging from ~1500 K in cooling regions, and >2500 K in the reaction front and advected particles. The temperatures for the reaction front are lower than the adiabatic flame temperatures for the mixture which is likely due to the addition of polymer binders (which nominally have flame temperatures in the ~2000 K region) incomplete reactions, and radiative loss. Measured temperature of the



samples did not largely change with additions of CuO to the mixture, which is expected since the calculated adiabatic flame temperature changes only slightly with small additions of CuO (Figure 5-7, Figure 5-8, Figure 5-9). An analysis on the energy transferred by advecting particles will be presented in later sections based on the pyrometry images presented below.



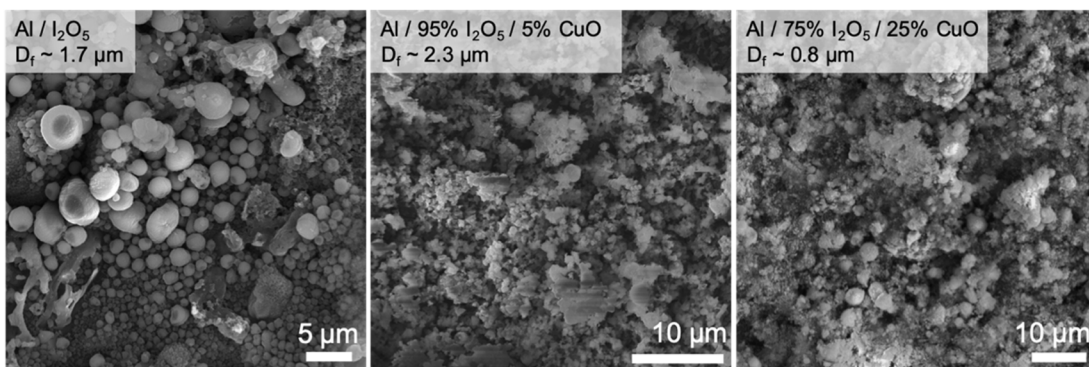
**Figure 5-2:** (a) High-speed microscopy images for the Al/75% I<sub>2</sub>O<sub>5</sub>/25% CuO sample and (b) corresponding temperature maps as calculated by color ratio pyrometry. (c) Drawing depicting different advection events that are observed in the reaction front for the high-speed microscopy videos. (d) Average particle velocity for particles that are ejected from the reaction front and example image for each mixture. Pictures in (d) have been brightened to clearly show streaks, but each picture has the same exposure time.

Velocities of the advected particles were estimated using the pixel/distance ratio for the assembly, estimated length of the particle path streak, and the exposure time of each frame. Despite the expectation that gas generation would increase the average advected particle velocity, the opposite phenomenon was found. With increasing CuO content, (see Figure 5-2d) the average particle velocity increases from ~3 m/s to ~18 m/s, both of which are ~100x faster than the bulk propagation speed. Assuming solid-to-gas conversion at constant pressure, the density difference should result in a factor of 1000x in gas velocity relative to the propagating front, in the absence of area expansion. Thus, the changes in particle velocity may result from a complex amalgam of gas production and expansion leading to particle advection. The main point, however, is that particle advection is clearly observed, consistent with the analysis of Egan et al., and will be discussed further in this paper.

#### 5.4.4 Post-reaction forensics

A major detractor from using a metal oxidizer such as CuO in nanoenergetic composites with Al is the tendency for materials to reactively sinter to form  $\mu\text{m}$ -sized particles that can effectively slow reaction rates and lead to incomplete combustion.<sup>47,166</sup> Reactive sintering has been observed in other high-speed microscopy studies that we have performed, however it has been proposed in other works that the addition of a gas generator can reduce the degree of reactive sintering.<sup>73,166</sup> In principle,  $\text{I}_2\text{O}_5$  would represent the ideal oxidizer since there is less metal that would be able to reactively sinter with the nAl and the constant gas generation would likely enhance particle dispersion, leading to smaller product particles.

To investigate the degree of reactive sintering in the CuO-doped Al/I<sub>2</sub>O<sub>5</sub> energetic composites, reaction products were imaged under a scanning electron microscope. Figure 5-3 shows the different particle morphologies for the reaction products as a function of CuO content. Interestingly, the rough final particle size did not appreciably vary from mixture to mixture, however, the final particle sizes were an order of magnitude smaller than those observed in a pure Al/CuO high-loading composite burn stick (~20μm).<sup>166</sup> This result suggests that the gas generated from the decomposition of I<sub>2</sub>O<sub>5</sub> significantly impacts the degree of reactive sintering, as expected, and that small additions of CuO do not significantly alter the final particle size. Particle morphology did seemingly change from mixture to mixture though, whereby the pure Al/I<sub>2</sub>O<sub>5</sub> had an overwhelmingly spherical particle shape and the morphology shifted towards the previously-reported “snowman” shape that would arise from nanoparticles sintering.<sup>47,114</sup> The apparent surface roughness of the CuO-doped samples is also comparable to the post-combustion product images reported by Jacob et al. where the small surface features are Cu vapor condensate.<sup>99</sup> The presence of these small Cu surface features suggests that the combined nucleation and cooling time of small particles is on the time scale of the propagation.<sup>99</sup> It could be reasonably expected that the degree of reactive sintering would steadily increase as a function of CuO content since the amount of gas generated would decrease. Despite this proposed behavior, the Al/I<sub>2</sub>O<sub>5</sub>/CuO mixtures explored in this study do not appear to significantly impact the overall degree of reactive sintering since the I<sub>2</sub>O<sub>5</sub> content is still relatively high.



**Figure 5-3:** Post-reaction SEM images for different CuO-doped Al/I<sub>2</sub>O<sub>5</sub> composites and final particle diameter ( $D_f$ ).

### 5.5 Discussion

The results show that small additions of CuO can dramatically increase the reliability of a high-loading Al/I<sub>2</sub>O<sub>5</sub> composite film. This observation points to the interplay of different heat transfer mechanisms on propagation that had previously only been estimated using back-of-the-envelope calculations. The discussion below explores the impact that advection and metal condensation have for sustained combustion in heterogeneous composites and proposes principles that can be extended to comparable systems.

Egan et al. estimated the extent to which the different heat transfer mechanisms contribute to the propagation of a tightly-packed Al/CuO composite, and has been adapted for the Al/I<sub>2</sub>O<sub>5</sub>/CuO composite system using the data presented in Table 5-1. The amount of energy required per unit length (E/mm) to sustain propagation was estimated with data from the NIST Webbook by calculating the amount of energy required to raise the temperature of Al (including the Al<sub>2</sub>O<sub>3</sub> shell) from room temperature to the ignition temperature of the Al/I<sub>2</sub>O<sub>5</sub> mixture (~860 K), the amount of energy required to raise the polymer composites to their decomposition temperature (~683 K)<sup>34</sup>, and the amount of energy required to decompose I<sub>2</sub>O<sub>5</sub> (~440 J/g measured

using TGA/DSC, see Figure 5-10). It was assumed that 10% of the mass of  $\text{I}_2\text{O}_5$  was actually water.<sup>169</sup> Since the Al/CuO ignition temperature ( $\sim 1050$  K) is higher than that for Al/ $\text{I}_2\text{O}_5$ , it was assumed local regions would ignite first, leading to ignition of neighboring Al/CuO and therefore the ignition temperature of Al/ $\text{I}_2\text{O}_5$  was used for all cases.<sup>11</sup>

Using the parameters in Table 5-3, it was calculated that a 2 mm-diameter packed tube would require  $\sim 1.9$  J/mm to sustain propagation for each of the samples – approximately 90% of the amount of energy required to ignite a printed Al/CuO burn stick with the same characteristics.<sup>11</sup> One expects that higher reaction temperature and reduction in reactive sintering would favor sustained combustion in the Al/ $\text{I}_2\text{O}_5$  system, but the opposite effects were observed. To investigate the root causes of this observation, a heat transfer analysis was employed for the various contributing factors that drive propagation, analogous to Egan et al. A summary of the data discussed below is shown in Figure 5-4.

#### 5.5.1 Conductive heat transfer estimates

Heat transfer from conduction was estimated using the same methods discussed in Egan et al. with slight modifications to the heat flux based on thermal measurements performed and estimated values of thermal conductivity. Thermal maps presented in Figure 5-2a show a temperature drop of  $\sim 1500$  K over a  $200\ \mu\text{m}$  distance, therefore the thermal gradient was estimated to be  $7.5 \times 10^6$  K/m for all of the samples. The thermal conductivity for the composite was estimated to be  $\sim 1.2$  W/mK based on a study by Zhou et al. where the thermal conductivity of Al/PVDF composites was measured as a function of volume fraction.<sup>170</sup> It is important to note that the thermal conductivity of

the composite material is much lower than that of pure Al because of thermal boundary resistance at the interface between Al particles and the polymer matrix.<sup>170</sup> Additionally, the area over which the heat could be transferred in the condensed medium was corrected using a 67% porosity estimate. Thus, from the measured thermal gradient and estimated material properties, the maximum heat flux from conduction should be  $< \sim 9 \text{ W}$  (Equation 5-3). This energy must be conducted in the amount of time that it takes for the reaction front to propagate a given distance, which in this case is less than 10 cm/s (Figure 5-1). Using this criterion, heat transfer via conduction only contributes  $\sim 0.09 \text{ J/mm}$  – i.e. less than 5% of what would be theoretically required. This is expected to be an upper estimate given that sample porosity will only lower the conductive heat flux. Therefore, it was concluded that thermal conductivity is at best a minor heat flux source.

#### 5.5.2 Radiative heat transfer estimates

Radiative heat transfer from hot particles back to the surface was estimated by assuming that the reaction would have peak temperatures at the isobaric adiabatic flame temperature and that particles generated from the reaction had an  $\epsilon=0.8$ .<sup>19</sup> Given the small area of condensed material being heated, this yielded a power of  $\sim 9\text{-}10 \text{ W}$  (Equation 5-4), however, the propagation rate of the material once again limited the amount of time that the radiation would be able to occur. The energy contribution from radiation was therefore only  $\sim 0.10 \text{ J/mm}$  – approximately 5% of the energy required.

### 5.5.3 Convective heat transfer estimates

Convective heat transfer in the system was assumed to have perfect efficiency driven by the transfer of energy from hot gases to the unreacted material. The amount of gas generated per unit length of material was determined by calculating how much  $I_2$  and  $O_2$  gas could be produced from the  $I_2O_5$  present in that length, and assuming that only these gases contributed to convective heat transfer.

Interestingly, the large quantity of gas generated during reaction is able to theoretically contribute  $\sim 3.2$  J/mm of reactant – more than 170% of what is required for the mixture to sustain propagation (Equation 6-5). This result suggests that, given a confined volume where all of the energy generated by the gas could be transferred back into the system (such as a pressure cell or burn tube), the  $Al/I_2O_5$  system would consistently be able to combust, however it is not a valid assumption for the system under consideration. For the constant pressure, open-environment burn stick experiments, the hot gas generated will also be driven away from the direction of propagation, as observed experimentally.<sup>35</sup>

As an estimate for the amount of mass, and therefore energy, that would be transferred toward the propagating front, it was assumed that the mass transfer roughly scales with porosity of the sample in accordance with Darcy's Law. If the pressure drop across the entire sample was equivalent to the pressure drop at lengths far away from the front, then the ratio of forward flow would scale as shown in Equation 5-1 where  $\Phi$  is porosity,  $Q_f$  is forward flow, and  $Q_{tot}$  is total flow. The samples were estimated to be 67% porous, therefore the fraction of forward flow is  $\sim 40\%$  of the total flow.<sup>157</sup> This ratio was used to estimate the percentage of the theoretical heat transfer would be

contributed toward propagation. This leads to an energy contribution from convection of ~1.2-1.3 J/mm – approximately 60-70% of the energy required.

$$\frac{Q_f}{Q_{tot}} = \frac{\Phi}{1 + \Phi} \quad (5 - 1)$$

#### 5.5.4 Metal vapor condensation estimates

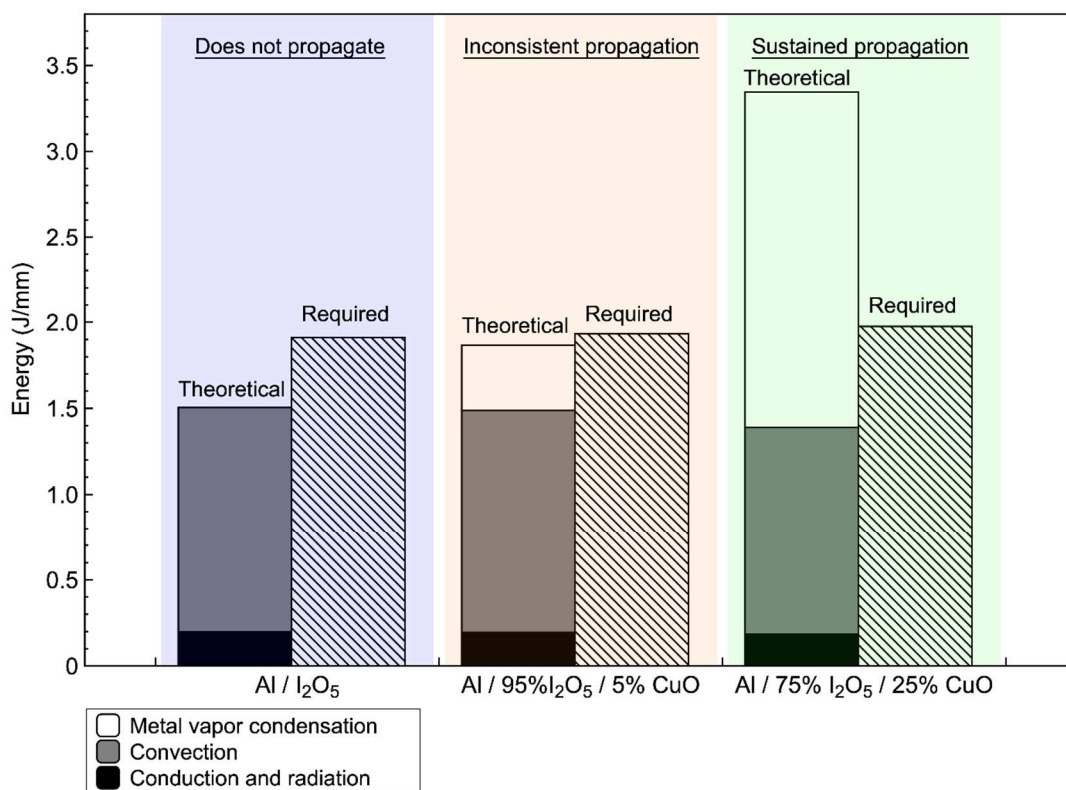
A major contributor to the heat transfer for propagation in Egan et al. was the condensation of metal vapors produced from the reaction. Equilibrium calculations performed in CHEETAH estimate the amount of condensable metal vapor in each mixture, other than Al, since oxidation is primarily expected to occur in the molten state. However, the amount of metal vapor that was allowed to condense was scaled using the same factor as was used in the convective heat transfer estimate since the vapor would flow with the other generated gases.

A major distinction between the analysis presented here and those in Egan et al. is the condition under which equilibrium calculations are performed. Since the experiment being modeled here is an open environment burn stick, equilibrium values for Cu vapor concentration correspond to constant pressure conditions. Egan et al. employed a constant volume condition since burn tube experiments are effectively a constant volume system.<sup>11</sup> The change to a constant pressure system actually favors the formation of Cu in the vapor due to the volume expansion despite the lower adiabatic temperature.<sup>9</sup> For thoroughness, the Al/CuO case was recalculated using values from a constant pressure equilibrium concentration of Cu and have presented the results in Figure 5-11.<sup>65</sup>



In the case of pure Al/I<sub>2</sub>O<sub>5</sub>, the lack of energy transfer from metal condensation dramatically impacts the amount of energy being transferred back to the unreacted material. Equilibrium calculations for the Al/I<sub>2</sub>O<sub>5</sub>/CuO find that ~0.5 mol and ~2.7 mol Cu vapor were produced per kg of reactant in the 5% and 25% CuO cases, respectively. Using heat capacity, heat of vaporization, heat of formation for Cu (NIST webbook), the Darcy Law correction to mass transfer (described above), and estimating that the vapor is cooling down from the isobaric adiabatic flame temperatures to the ignition temperature, ~0.38 J/mm (20% of required) and ~1.95 J/mm (99% of required) could be transferred to unreacted material for the 5% and 25% CuO cases, respectively (Equation 5-6).

It is important to consider the role that metal vapor condensation plays in propagation since the small particle condensates are clearly evident in the final product morphology of samples that had CuO. This heat would be transferred in a relevant time scale since Jacob et al. showed that the combined nucleation and cooling time of metal vapor is on the order of microseconds.<sup>99</sup> This suggests that even small additions of oxidizer with a condensable metal vapor dramatically reduces the reliance on advection for sustainable combustion and could even be eliminated. These calculations are consistent with the observation that CuO additives are necessary in order for Al/ I<sub>2</sub>O<sub>5</sub> high loading systems to propagate.



**Figure 5-4:** Estimated values for the theoretical energy transfer and the amount of energy required to sustain combustion for different mixtures of Al/I<sub>2</sub>O<sub>5</sub>/CuO.

#### 5.5.5 Advective heat transfer

Advective heat transfer of hot particles to unreacted areas is very evident in the high-speed microscopy shown in Figure 5-2. Egan et. al simply estimated the amount of advective heat transfer based on the heat that could be recovered from molten Al and Cu particles, however, the high-speed pyrometry/microscopy apparatus can be used to estimate the amount of energy that is transferred.

The energy content of each particle ( $E_p$ ) is estimated by Equation 5-2 where the thermophysical properties (density:  $\rho_p$ , heat capacity:  $C_p$ ) of the particle are assumed to be that for Al<sub>2</sub>O<sub>3</sub> and the particle volume ( $V_p$ ) can be calculated from the final particle size found in post-forensic SEM (~1-5 $\mu$ m). Assuming that the particle cools from the

measured particle temperature ( $\sim 3000$  K) to the ignition temperature, each particle carries  $\sim 0.1$   $\mu\text{J}$  (assuming a  $2.5$   $\mu\text{m}$ -diameter particle). Therefore, the number of particles that need to be generated per unit length of material can be calculated by dividing the energy gap by  $E_p$ . The number of particles required to make up the energy deficit, assuming that convection makes no contribution to the energy transported and all of the condensed metal vapor is transferred back to the film, is  $\sim 4.4 \times 10^6$  particles/mm and  $\sim 7.2 \times 10^5$  particles/mm for Al/I<sub>2</sub>O<sub>5</sub> and Al/95% I<sub>2</sub>O<sub>5</sub>/5% CuO composites, respectively. Thus a 5% addition of Cu results in an 83% reduction in required particle advection and highlights the importance of metal vapor condensation on reactive material propagation.

$$E_p = \rho_p V_p C_p (T_{\text{ad}} - T_{\text{ign}}) \quad (5 - 2)$$

The simple analysis illustrates the fine balance between the different modes of heat transfer for propagation in solid combustible materials. In scenarios where reliable propagation is critical, such as a stable burn of a solid propellant or delivery of a biocide, small additions of an oxidizer with a condensable metal vapor could be the difference between guaranteed performance and critical failure.

## 5.6 Conclusions

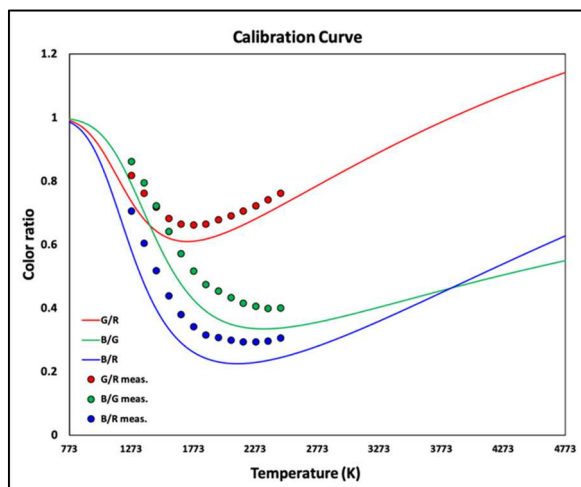
Reactive sintering can negatively impact reactions in energetic composites prepared with nanoparticles since the loss of nanostructure can decrease burn rates and prematurely arrest reactions. Iodine pentoxide is an ideal oxidizer to reduce the amount of reactive sintering since the overwhelming gas generation will effectively atomize any sintering particles and it has immediate applications as a biocidal additive. When investigating high-loading Al/I<sub>2</sub>O<sub>5</sub> films, the burn sticks were unable to propagate.

Small additions of CuO (5-25%) to the Al/I<sub>2</sub>O<sub>5</sub> mixture increased reliability in propagation with CuO content. High-speed pyrometry and microscopy were used to estimate advective heat transfer in the system and post-products showed only minor variation in the degree of reactive sintering with CuO addition. A simple energy balance performed to estimate the composite's reliance on advective heat transfer for propagation found that the elimination of condensing metal vapor (Cu from CuO) dramatically increased this dependency. It is proposed that small additions of CuO, resulting in a condensable product vapor, can effectively eliminate the reliance on heat transfer via advection enabling a more consistent propagation. These results highlight the fine balance between different heat transfer mechanisms that drive propagation in reactive materials.

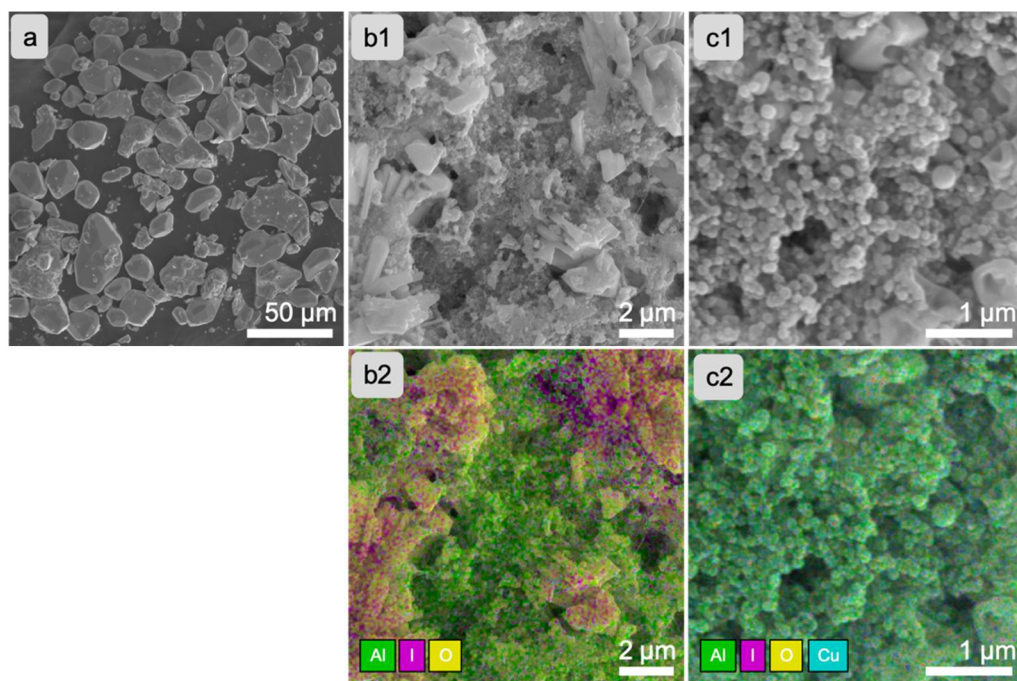
### ***5.7 Acknowledgements***

The authors gratefully acknowledge support from the Defense Threat Reduction Agency and the Air Force Office of Scientific Research. The authors would also like to acknowledge B. Aaron Mason (NAWS-CL) for his assistance in calibrating the K2 Distamax microscope assembly. Electron microscopy was performed using a FEI NNS450 SEM in the Central Facility for Advanced Microscopy and Microanalysis at University of California, Riverside. Reproduced from D.J. Kline\*, Z. Alibay\*, M.C. Rehwoldt, A. Idrogo-Lam, S.G. Hamilton, P. Biswas, F. Xu, M.R. Zachariah, Experimental observation of the heat transfer mechanisms that drive propagation in additively manufactured energetic materials, *Combust. Flame* 215 (2020) 417-424. doi:10.1016/j.combustflame.2020.01.020 with permission from Elsevier. \*Denotes that authors contributed equally to this work.

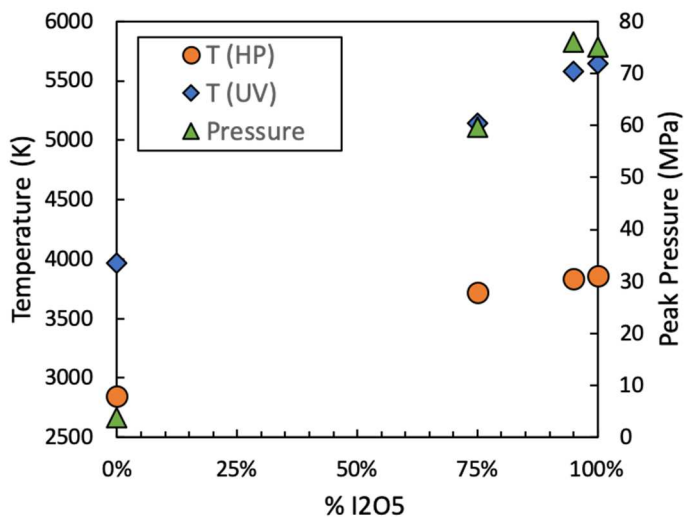
## 5.8 Supplemental Information



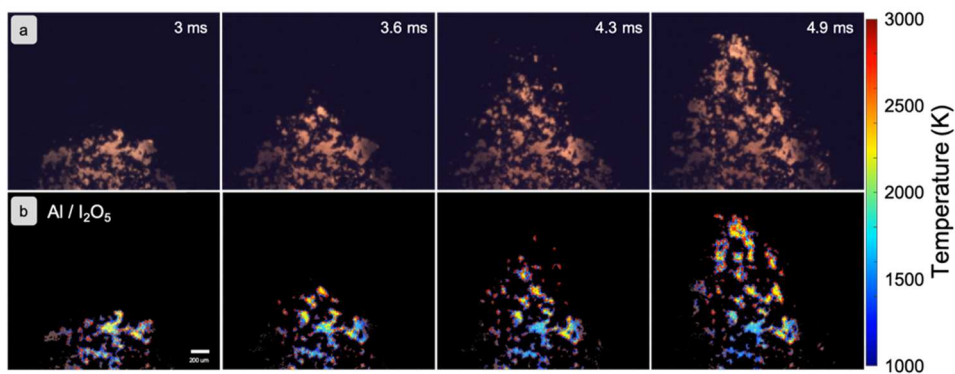
**Figure 5-5:** Theoretical color channel ratios vs. temperature for the Phantom VEO710L and measured color ratios for the camera and lens assembly.



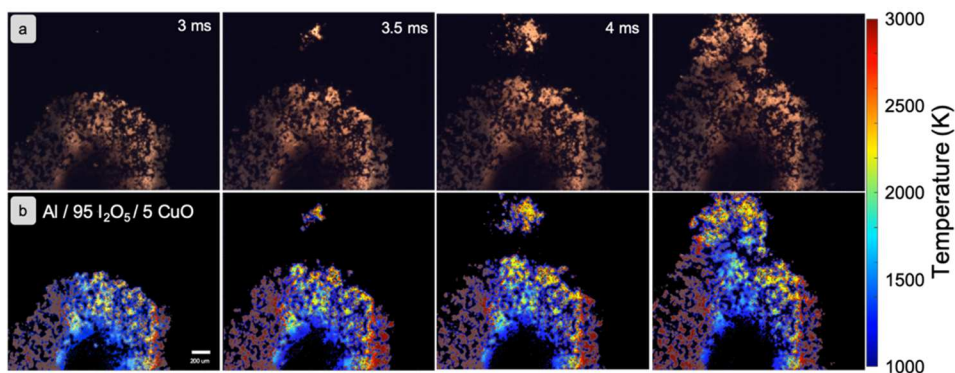
**Figure 5-6:** (a) Commercially purchased  $I_2O_5$  crystals. (b1) SEM image of the  $Al/I_2O_5$  high loading burn stick and (b2) corresponding element map for the sample. (c1) SEM Image of the  $Al/75\% I_2O_5/25\% CuO$  high loading burn stick and (c2) corresponding element map for the sample.



**Figure 5-7:** Constant pressure (T HP) and constant volume (T UV) adiabatic flame temperature and equilibrium pressure for constant volume combustion. Constant volume calculations assume a 6% theoretical maximum density.



**Figure 5-8:** (a) High-speed microscopy images for the Al/I<sub>2</sub>O<sub>5</sub> sample and (b) corresponding temperature maps as calculated by color ratio pyrometry.



**Figure 5-9:** (a) High-speed microscopy images for the Al/95% I<sub>2</sub>O<sub>5</sub>/CuO sample and (b) corresponding temperature maps as calculated by color ratio pyrometry.

**Table 5-2:** Oxygen release temperatures for different oxidizers as measured using temperature-jump time-of-flight mass spectrometry.

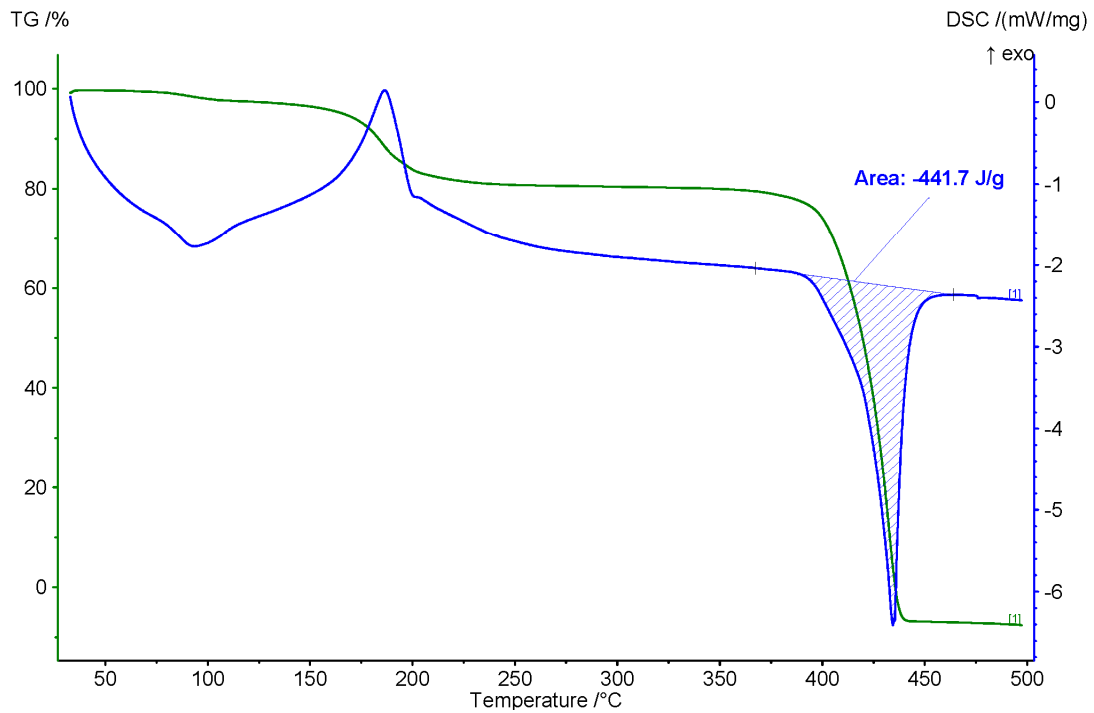
Material	O <sub>2</sub> release T (C)
I <sub>2</sub> O <sub>5</sub>	421
I <sub>2</sub> O <sub>5</sub> /CuO	476
CuO	693

**Table 5-3:** Parameters for estimating heat transfer requirements in Al/I<sub>2</sub>O<sub>5</sub> system for propagation.

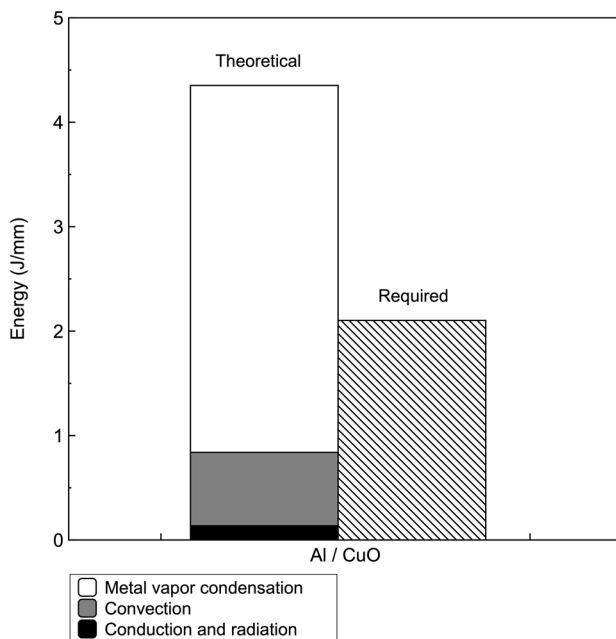
Material System	wt.% Polymer	Tube Diam.	% TMD	Ign. T	Mass/mm	E/mm
Al / I <sub>2</sub> O <sub>5</sub>	4% PVDF, 6% HPMC	2 mm	33%	863 K	0.43 mg/mm	1.91 J/mm
Al / 95% I <sub>2</sub> O <sub>5</sub> / 5% CuO	4% PVDF, 6% HPMC	2 mm	33%	863 K	0.44 mg/mm	1.93 J/mm
Al / 75% I <sub>2</sub> O <sub>5</sub> / 25% CuO	4% PVDF, 6% HPMC	2 mm	33%	863 K	0.46 mg/mm	1.98 J/mm

**Table 5-4:** Measured and theoretical densities of the Al/I<sub>2</sub>O<sub>5</sub> composites with calculated porosity values.

Material System	Measured density (g/cm <sup>3</sup> )	Theoretical density (g/cm <sup>3</sup> )	Porosity (%)
Al / I <sub>2</sub> O <sub>5</sub>	1.56	4.12	62
Al / 95% I <sub>2</sub> O <sub>5</sub> / 5% CuO	1.63	4.17	61
Al / 75% I <sub>2</sub> O <sub>5</sub> / 25% CuO	1.79	4.39	59



**Figure 5-10:** Thermogravimetric analysis/differential scanning calorimetry results for  $I_2O_5$ .



**Figure 5-11:** Estimated values for the theoretical energy transfer and the amount of energy required to sustain combustion for Al/CuO using constant pressure equilibrium calculations. Propagation velocity assumed to be 802 m/s.<sup>65</sup>

$$E_{cond} = \frac{kA_{tube}(1 - \Phi)}{v_{prop}} \left( \frac{dT}{dx} \right) \quad (5 - 3)$$



$$E_{rad} = \frac{\sigma \varepsilon (1 - \Phi) A_{tube} (T_{AFT} - T_0)}{v_{prop}} \quad (5 - 4)$$

$$E_{conv} = \left( \frac{Q_f}{Q_{tot}} \right) \left( \frac{m_{I_2}}{L} \int_{T_{ign}}^{T_{AFT}} C_{p,I_2}(T) dT + \frac{m_{O_2}}{L} \int_{T_{ign}}^{T_{AFT}} C_{p,O_2}(T) dT \right) \quad (5 - 5)$$

$$E_{vap} = \left( \frac{Q_f}{Q_{tot}} \right) \frac{m_{sample}}{L} m_{Cu}^* \left( \int_{T_b}^{T_{AFT}} C_{p,Cu}(T) dT + \Delta H_{vap,Cu} + \int_{T_m}^{T_b} C_{p,Cu}(T) dT + \Delta H_{f,Cu} + \int_{T_{ign}}^{T_b} C_{p,Cu}(T) dT \right) \quad (5 - 6)$$

**Table 5-5:** Variables for equations 5-3 to 5-6.

Symbol	Name
k	Thermal conductivity (W/mK)
$\sigma$	Stefan-Boltzmann constant (W/m <sup>2</sup> K <sup>4</sup> )
$\varepsilon$	Emissivity
C <sub>p</sub>	Heat capacity (J/gK)
$\Delta H_{vap}$	Heat of vaporization (J/g)
$\Delta H_f$	Heat of formation (J/g)
T <sub>AFT</sub>	Adiabatic flame temperature (K)
T <sub>ign</sub>	Ignition temperature (K)
T <sub>b</sub>	Boiling temperature (K)
T <sub>0</sub>	Initial temperature
dT/dx	Thermal gradient (K/m)
m <sub>sample</sub>	Sample mass (g)
m <sub>I<sub>2</sub></sub>	Mass of I <sub>2</sub> (g)
m <sub>O<sub>2</sub></sub>	Mass of O <sub>2</sub> (g)
m* <sub>Cu</sub>	Mass of Cu vapor per kg reactant at equilibrium as calculated using Cheetah (g)
Q <sub>f</sub> /Q <sub>tot</sub>	Ratio of forward mass transfer to total transfer (estimated using Equation 5-1).
$\Phi$	Porosity of sample
v <sub>prop</sub>	propagation velocity (m/s)
A <sub>tube</sub>	Cross-sectional area of tube (m <sup>2</sup> )
L	Sample length (m)

## Chapter 6: *In-operando* thermophysical properties and kinetics measurements of Al-Zr-C composites

### 6.1 *Summary*

This work investigates the combustion velocity, thermophysical properties, and reaction activation energies of Al-Zr-C nanolayered composite microparticles undergoing high-temperature synthesis after preparation via additive manufacturing. High-speed videography and pyrometry of the reaction event were used to analyze two Al-Zr-C samples with varied stoichiometry. Combustion velocity of the Al-Zr-C composites was  $\sim 0.3\text{-}0.5$  cm/s and varied inversely with the Al content in the system. The Al-Zr-C composites also exhibited auto-oscillations during the propagation event which were characterized to have temperature fluctuations of  $\sim 50\text{-}100$  K with a periodicity of  $\sim 1$  Hz. Temperature data collected via color ratio pyrometry was used to measure the thermal profile in-operando. Temperature maps were used to estimate the thermal diffusivity of the samples to be  $\sim 2 \times 10^{-6}$  m<sup>2</sup>/s on the leading edge of the reaction front with a  $>30\times$  increase in thermal diffusivity on the trailing edge. The activation energy for the Al-Zr-C composites was estimated to be  $\sim 30\text{-}35$  kJ/mol under reacting conditions. This work ultimately demonstrates an accessible measurement methodology that could be used to estimate thermophysical changes in materials for generalized modeling purposes and confirms the functionality of the materials to create conductive pathways after reaction.

### 6.2 *Introduction*

Energetic materials from the perspective of self-propagating high-temperature synthesis (SHS) systems have experienced a revived research interest with the rise of new manufacturing methods<sup>35,78,86,90</sup> and new experimental characterization techniques (high-speed videography/microscopy).<sup>73,166,171</sup> Many efforts in the literature have investigated structure-function relationships by studying changes in the chemical formulation<sup>34,54,73</sup> or by changing the physical architecture of the samples using different manufacturing methods<sup>86,87</sup>, material properties<sup>91,172,173</sup>, or additives that may alter propagation rates.<sup>95,174–176</sup> These studies have, for the most part, focused on the behavior of thermite systems (metal fuel/metal oxide oxidizer), which are a subset of SHS systems.<sup>177</sup> Comparison studies on these materials have attempted to independently modify thermochemical and architectural parameters, though they are inherently entangled. For example, changes in stoichiometry may change chemistry and also change overall thermophysical properties.<sup>95,174</sup> The multiphase nature of energetic material reactions only further complicates the process of making reliable thermophysical measurements.

In contrast to many energetic reactions where fast pressurization and energy release occurs, the self-propagating high-temperature synthesis process can be “gasless,” achieve similar temperatures, and nearly all final products are condensed phase since the process is typically an exothermic metal alloying process.<sup>178</sup> SHS has been widely studied since the early 1970s as a method to produce powders or ceramics. Materials for SHS are generally mixtures of micron-sized metal powders that have been mechanically milled into composite materials that will react and propagate at ~1 cm/s.<sup>178–180</sup> Significant strides have been made by others to develop models for the heat

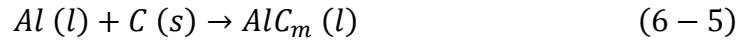
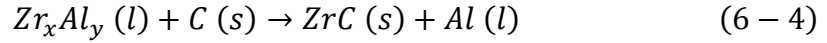
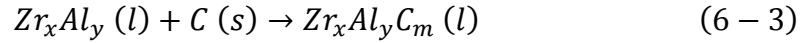
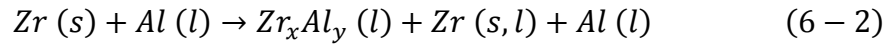
transfer and kinetics of these reactions.<sup>179,181,182</sup> Although some SHS composites prepared by arrested reactive milling release gas, for this study, the condensed phase nature of intermetallic SHS reactions are favorable for a diagnostic case study.<sup>183,184</sup>

A fundamental aspect of heterogenous, solid, self-sustained interface driven reactions is the interplay between the heat flux to initiate the reaction and the heat generated from reaction. At a macroscopic level, nearly all SHS systems act in a “quasi-homogenous” manner whereby the reaction front of the SHS system moves uniformly in the direction of propagation.<sup>177,185</sup> This can simply described by saying that the heat flux smoothly delivers heat forward for initiation to nearby unreacted materials.<sup>177,186,187</sup> Steady-state, quasi-homogenous SHS systems may be quantified analytically by Equation 6-1 where  $v$  is propagation velocity,  $\alpha$  is effective thermal diffusivity,  $\dot{\omega}$  is reaction rate,  $T_{ad}$  is adiabatic flame temperature,  $T_i$  is ignition temperature, and  $T_0$  is the temperature of the unreacted material.<sup>187</sup> Equation 6-1 simply suggests that the propagation rate is dependent on how much energy is released ( $T_{ad}$ ), how fast the energy is released ( $\dot{\omega}$ ), and how fast that energy can be transported to nearby unreacted reactions ( $\alpha$ ).

$$v \approx \sqrt{\alpha \cdot \dot{\omega} \frac{(T_{ad} - T_i)}{(T_i - T_o)}} \quad (6 - 1)$$

This study will focus on the self-propagating, high-temperature synthesis of Al-Zr-C nanolayer composites prepared by arrested reactive ball milling and subsequently 3D printed. This mixture has particular applications as a composite that can be reacted to form electrically conductive paths as detailed in concurrently-prepared articles by Arlington et al.<sup>188,189</sup> and previously patented by Fritz and Barron.<sup>190</sup> Al-Zr-C

composites have been largely studied prior to this for ZrC particle synthesis, but much of the research has only been reported by one group and reiterated numerous times.<sup>180,191–193</sup> Previous work reports that the reaction is initiated by Al melting (~933 K) followed by an exothermic alloying process with Zr to form  $Zr_xAl_y$  (Eq. 6-2).<sup>180,191–194</sup> The C then dissolves into the mixture to form a slurry of  $Zr_xAl_yC_m$  (Eq. 6-3) until a saturation point where ZrC particles begin nucleating (Eq. 6-4). The final products of the SHS reaction are mixtures of Al, Zr, and  $Zr_xAl_y$  alloys with small ZrC particles interspersed throughout. It has also been shown that some  $AlC_m$  can be found in the final products (Eq. 6-5).<sup>180,191–193</sup> The ideal overall reaction for the Al-Zr-C system is shown in Equation 6-6 as the ZrC formation reaction, in which  $\Delta H_{rxn} = -196.6$  kJ/mol.



Research on SHS materials has largely relied on measurements taken with highly specialized diagnostics like time-resolved X-ray diffraction (TRXRD), high-speed transmission electron microscopy (HSTEM), or dynamic transmission electron microscopy (DTEM), though these tools can be prohibitively expensive and time intensive despite offering *in-operando* observations.<sup>179</sup> These *in-operando* devices also typically involve experiments that are performed under nonstandard conditions (e.g. in vacuum) and on small amounts of material and generally do not offer simultaneous temperature measurements for thermophysical property and kinetics estimations.

However, others have found success measuring the thermophysical properties and kinetic parameters of SHS reactions using temperature measurements which has served as the inspiration for the work here.<sup>179,195,196</sup> This study observes and quantifies flame front behavior through an *in-operando* probe of the thermophysical and kinetic properties of an Al-Zr-C SHS material using high-speed color camera pyrometer. Thermochemical parameters can be used as inputs for a simple thermal reaction propagation models for predictive modeling of propagation rates in SHS materials based on easily measured input parameters. Additionally, the *in-operando* spatiotemporal temperature measurements of the SHS reaction will be used to gain insight on the reaction mechanism in the material.

### **6.3 Methods**

#### **6.3.1 Sample preparation**

The powder samples were prepared by Charles Stark Draper Laboratory (CSDL) (Cambridge, MA, USA) and Johns Hopkins University (Baltimore, MD, USA) as part of the RAIL3D program and previously patented as “Reactively assisted ink for printed electronic circuits.”<sup>190</sup> A full characterization of these powders is described in concurrently prepared articles by Arlington et al.<sup>188,189</sup> In summary, these powders were prepared by arrested reactive ball milling. The mass loading of the powders was chosen to give an Al:Zr:C atomic ratio of 2:1.2:1 or 0.5:1.2:1. After milling, powders were dried in air and sieved to less than 25 micrometers. The composition of the two tested samples can be seen in Table 6-1.

**Table 6-1:** Mixture compositions for the two samples tested in this study. Number in sample description denotes atom ratio of reactants in composite.

Sample	Mass %		
	Al	Zr	C
<b>2 Al / 1.2 Zr / 1 C</b>	31%	62%	7%
<b>0.5 Al / 1.2 Zr / 1 C</b>	10%	81%	9%

Samples used in this study were prepared by direct-write additive manufacturing (3D printing) of an ink containing one of two different stoichiometries as described in Arlington et al.<sup>188,189</sup> Prior to printing, the precursor was loaded into a 10 mL plastic syringe with a tapered 18-gauge blunt Luer Lock needle (McMaster-Carr) and placed into a vertically mounted syringe pump on the 3D printer (Hyrel3D). The syringe pump extrudes material along an 8-cm square perimeter path that has 12 microscope coverslips (22x22mm, 0.17mm thickness, VWR) placed along the path. Four layers of material was printed for each batch of 12 samples for a final approximate height of ~2.5-3 mm. Samples were then placed on a hot plate at ~100C for ~10 minutes to evaporate any remaining DMF from the samples and are then broken apart using a razor. Samples were printed to this height to minimize the effect of thermal loss through interactions with the glass substrate on which they were mounted that could possibly skew values for temperature and propagation rate.

### 6.3.2 Morphology characterization

Reactive powders and printed samples were analyzed using scanning electron microscopy (SEM) using a Duobeam Quanta 200i with SEM pictures shown in Figure 6-5. SEM analysis of the Al-Zr-C system focuses on the particle separation distance, particle sizes, and particle morphology. From Figure 6-5, one can see that reactant particles within both systems considered are mostly non-spherical with length scales

which appeared to be  $< 10\text{-}20\ \mu\text{m}$  as a result of the ball milling process. Separation distances appeared to be no larger than a couple microns which seems reasonable given that the mode of additive manufacturing relies on the evaporation of a solvent, resulting in materials which are expected to be quite porous.

### 6.3.3 *High-speed videography/pyrometry*

Macroscale burn tests were performed by mounting the microscope coverslips with printed SHS material on a 3-axis translational stage (Newport) inside of a container filled with argon and observing the combustion event using high-speed videography. Samples were ignited using a resistively heated nichrome wire and the reaction event was recorded using a high-speed color camera (Phantom VEO710) recording at  $\sim 200$  frames/s. The camera was equipped with a macro lens that allows for a spatial resolution of  $\sim 20\ \mu\text{m}/\text{pixel}$  (Nikon). The propagation velocity for each sample was estimated by tracking the location of the reaction front throughout the video and using the pixel/distance ratio to calculate distance traveled over time. Samples were tested three times each and propagation rate was averaged over each of the videos using a linear best fit. A separate experiment to observe the homogeneity of the reaction front used the same high-speed camera equipped with a long working distance microscope objective which gives a spatial resolution of  $\sim 1.7\ \mu\text{m}/\text{pixel}$  (InfinityOptics K2 DistaMax).

The high-speed color camera was also calibrated to operate as a pyrometer enabling both temporal and spatial temperature measurement of the reacting samples. This capability has been described in detail in previous work.<sup>73</sup> Briefly, raw pixel values were extracted from raw images and demosaiced in accordance with the



camera's Bayer filter array to recover red, green, and blue color intensities at every pixel.<sup>73</sup> The color channel ratios were then calculated and corrected using calibration factors obtained for the camera using a blackbody radiation source.<sup>73</sup> Corrected color ratios were then matched to a temperature assuming gray body emission. The final temperatures reported in the 2D temperature map are below the saturation threshold, above the black level of the camera, and are accurate to within  $\pm 110$  K.<sup>73</sup>

As will be discussed in the following sections, it was necessary to extend the dynamic range of the temperature measurements in order to extract information below the limit of sensitivity of our color ratio pyrometry method. Since light emission scales as a power-law of temperature, the pyrometry can be used to calibrate the raw light emission signal, thus enabling us to extend the temperature measure to lower temperatures. An example of the fit between the light emission-based temperature and color-ratio pyrometry estimated temperature can be seen in Figure 6-2.

#### 6.3.4 Determination of transport and kinetic properties

Simultaneous determination of transport and kinetic properties in condensed reaction/diffusion systems has been of long-standing interest and a variety of methods have been developed to estimate these parameters. For a detailed discussion on these methodologies as they are related to SHS, it is recommended that the readers reference the review article by Mukasyan et al.<sup>179</sup> The conversion rate ( $d\eta/dt$ ) of an SHS reaction can be represented by superposition of a reaction kinetic term that follows Arrhenius behavior,  $K(T)$  and a conversion term,  $\Phi(\eta)$  (Eq. 6-7).<sup>179</sup>

$$\frac{d\eta}{dt} = K(T)\Phi(\eta) \quad (6 - 7)$$

Previous reports by Boddington et. al suggested that the Arrhenius form of  $d\eta/dt$  can be represented by Equation 6-8, which consists of a pre-exponential factor (A), a conversion ( $\eta$ ) term raised to a factor  $n$  which is representative of “nth order kinetics,” and the Arrhenius exponential term where  $E_a$  is activation energy of the reaction,  $R$  is the ideal gas constant, and  $T$  is the temperature. It is important to note, however, that there are many potential kinetic rate laws which specifically apply to solid-state kinetics that could be used as an expression for  $d\eta/dt$ .<sup>197</sup> Boddington et. al reported that other forms of  $d\eta/dt$  were considered, but did not significantly change estimated values for A,  $n$ , and  $E_a$  or the quality of the fit. As such, other studies by have also used this expression to estimate kinetic parameters.<sup>196,198,199</sup>

$$\frac{d\eta}{dt} = A(1 - \eta)^n \exp\left(-\frac{E_a}{RT}\right) \quad (6 - 8)$$

Many methods have been developed to estimate  $E_a$  of SHS systems since it is fundamentally important to modeling reacting systems. One common method employed to estimate  $E_a$  involves dilution of the sample so that changes in combustion velocity are used to compute  $E_a$ .<sup>200</sup> However, considering that this method also inherently changes the chemistry of the reaction, an alternative proposed by Boddington et al. was favorable for the analysis presented here. Boddington et al. proposed that the conversion rate can be related to the reaction front temperature profile, thermal diffusivity, and velocity of the system by the Equation 6-9 where  $T_{ad}$  is the adiabatic flame temperature,  $T_0$  is the ambient temperature,  $t_r$  and  $t_d$  are the rise and decay times of the reaction in the remote inert regions of the reaction profile.<sup>179,195</sup>

$$\frac{d\eta}{dt} = \frac{\frac{T - T_0}{t_d - t_r} + \frac{dT}{dt} - \frac{\alpha}{v^2} \frac{d^2T}{dt^2}}{T_{ad} - T_0} \quad (6 - 9)$$

The adiabatic flame temperature for the ZrC formation reaction (Eq. 6-6) was calculated to be 3765 K using CHEETAH, though the actual temperature of the reaction would be reduced when considering side reactions with the Al.<sup>9,180,193</sup> Adiabatic flame temperature calculations for the 0.5 Al / 1.2 Zr / 1 C and 2 Al / 1.2 Zr / 1 C systems were estimated to be 2730 K and 1770 K, respectively, though these calculations are limited by the catalog of information available in the program for the Al-Zr-C intermediates.<sup>9</sup> Seeing that estimation of the adiabatic flame temperature with consideration for the Al content is difficult, the adiabatic flame temperature of the ZrC formation reaction was used for calculations to estimate activation energy in the sections below.

### 6.3.5 Thermophysical property estimations

Temperature measurements collected from color-ratio pyrometry and the light emission estimates discussed in the section above were used to extract thermophysical properties of the leading and trailing edges of the reaction. Prior to any calculations, the color-ratio pyrometry temperature values were smoothed using a 2D smoothing algorithm which utilized an 8-point smoothing window.<sup>201</sup> Single line temperatures were extracted using a custom routine on the post-processed data that allows for data visualization at a constant vertical position.

As discussed above, light emission-based temperature estimates were used to extend the dynamic range of our temperature measurements to much lower temperatures enabling us to probe the pre-reaction zone. After fitting the temperature

to light intensity, the extended temperature profile was analyzed and used as inputs to the Boddington equation (Equation 6-9). Assuming that reaction at low temperatures is negligible (pre-reaction zone; i.e.  $d\eta/dt = 0$ ), the measured combustion velocity and rise and fall time estimates from built-in MATLAB routines were used to extract values for the thermal diffusivity ( $\alpha$ ). Temperature profiles and derivative terms of the Boddington equation in the pre-reaction zone were estimated by fitting the light emission-based temperature measurements to exponential functions using built-in MATLAB fitting routines.

Thermal diffusivity of the post-reaction products was also of particular interest to examine the change in material properties after reaction. Given the gasless nature of the Al-Zr-C system, it can be assumed that heat transfer is governed exclusively by thermal conduction driven by thermal gradients.<sup>187,202</sup> The temperature of a sample as a function of position and time can therefore be roughly modeled by the heat diffusion equation (Equation 6-10) where  $T$  is temperature,  $x$  is position, and  $t$  is time. Estimations for thermal diffusivity can therefore be calculated using numerical differentiation on the pyrometry images recorded.

$$\frac{dT}{dt} = \alpha \left( \frac{d^2T}{dx^2} \right) \quad (6 - 10)$$

Initial attempts to estimate the thermal diffusivity of a sample by analyzing single point temperature as a function of time ( $dT/dt$ ) and using proximity temperatures to calculate ( $d^2T/dt^2$ ) numerically revealed that noise would generate unrealistic thermal diffusivities. Assuming a constant, quasi-homogeneous propagation rate, Equation 6-10 can be simplified by substituting  $x = vt$ , where  $v$  is propagation

rate (in m/s). Note that the propagation rate is shown to be very constant in Figure 6-1 and that the previous work on similar stoichiometries also shows a consistent propagation rate. Thermal diffusivity can therefore be calculated using Equation 6-11.

$$\frac{dT}{dx} = \frac{\alpha}{v} \left( \frac{d^2T}{dx^2} \right) \quad (6 - 11)$$

Temperature measurements used for estimating thermal diffusivity in the post reaction zone were extracted using color-ratio pyrometry values in a temperature region below the observed phase transition in the material (~1500 K). Data in this region was fit to an exponential function and those fits were then used to estimate the first and second derivatives of the thermal profile for Equation 6-11.

It is also important to assess the impact of other heat losses on the surface temperature and thermal diffusivity estimations. To estimate the impact of conductive heat loss to the glass substrate, a COMSOL Multiphysics simulation of the process was performed using standard values for thermal transport properties of materials and holding a hot “reaction zone” at ~1700 K. The thermal profiles of a system with and without the glass substrate revealed that the temperature only differed by <1% and therefore we have assumed that heat loss to the substrate was negligible for the time scale over which measurements were made. Results from the COMSOL simulation can be seen in Figure 6-6. A simple estimation of the relative heat fluxes due to convection and radiation also show that they play a minimal role relative to conduction in this region of interest. Both convective and radiative heat losses had an estimated flux which was <5% of that expected for conduction in the same region.

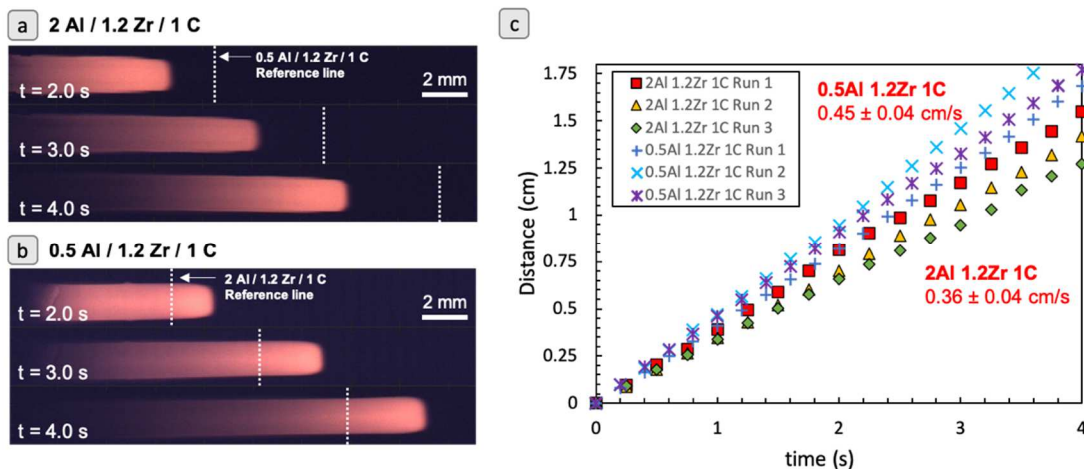
### 6.3.6 Kinetic parameter estimation

Kinetic rates were calculated using custom built routines in MATLAB using data that was extracted from the determination of the leading-edge thermal diffusivity measurements described above. By using the temperature values in the region on the leading edge above 933 K, the estimated rise and decay times, and velocity, values for  $d\eta/dt$  could be calculated using the Boddington equation (Eq. 6-9). The values estimated by the Boddington equation and measured temperature data were then used as inputs to a MATLAB solver routine which would minimize the difference between the values calculated in Equation 6-8 and Equation 6-9 to estimate values for the constants A, n, and  $E_a$ .<sup>195</sup>

## **6.4 Results**

### 6.4.1 Propagation characterization

The combustion event of samples was recorded using a high-speed camera with a resolution of  $\sim 20 \mu\text{m}/\text{pixel}$  and the leading edge of the reaction front was tracked as a function of time to calculate an average combustion velocity. Figure 6-1 shows position of the leading edge vs. time for the tested samples, all of which showed highly linear behavior indicating that mixing inhomogeneity from printing was not impacting the reaction velocity. Combustion velocities obtained were  $0.36 \pm 0.04 \text{ cm/s}$  and  $0.45 \pm 0.04 \text{ cm/s}$  for the 2 Al / 1.2 Zr / 1 C and 0.5 Al / 1.2 Zr / 1 C samples, respectively. This propagation rate is slightly lower than those reported by Vadchenko et al.<sup>194</sup> ( $\sim 1 \text{ cm/s}$ ), but this may be attributed to different preparation methods or heat sinks introduced by the glass substrate and polymer binder addition.



**Figure 6-1:** (a,b) Sample high-speed images of the propagating SHS samples tested. (c) Position vs. time data for combustion of 0.5 Al / 1.2 Zr / 1 C and 2 Al / 1.2 Zr / 1 C composites in Ar measured using high speed videography. Reported velocity is based on an average of linear fits between three experiments.

The inverse relationship between the Al content and combustion velocity is attributable to the change in thermal conductivity of the samples and a reduction in reaction completion as a function of Al content. A similar trend is also exhibited in work done by Arlington et al.<sup>188,189</sup> The Al in the SHS system is introduced to facilitate nanolayered particle fabrication, though a side effect of the addition is an increase in the thermal conductivity of the sample. However, the Al also results in side reactions with the Zr that would decrease the overall conversion for the ZrC formation reaction. Although it has been proposed that the propagation velocity would increase with the square root of thermal diffusivity and would thereby increase with Al addition, side reactions reduce the adiabatic flame temperature for the mixture and the increased thermal diffusivity reduce the peak temperature achieved in the composite. The combined effects of the increased thermal diffusivity and reduced reaction conversion after a ~3x increase in Al content actually reduced the propagation velocity in these experiments. To further probe the role of Al content reaction dynamics, color ratio

pyrometry on the propagation experiments is used in the following sections to estimate thermal profiles in-operando, thermal diffusivity of the composites, and activation energy of the reaction.

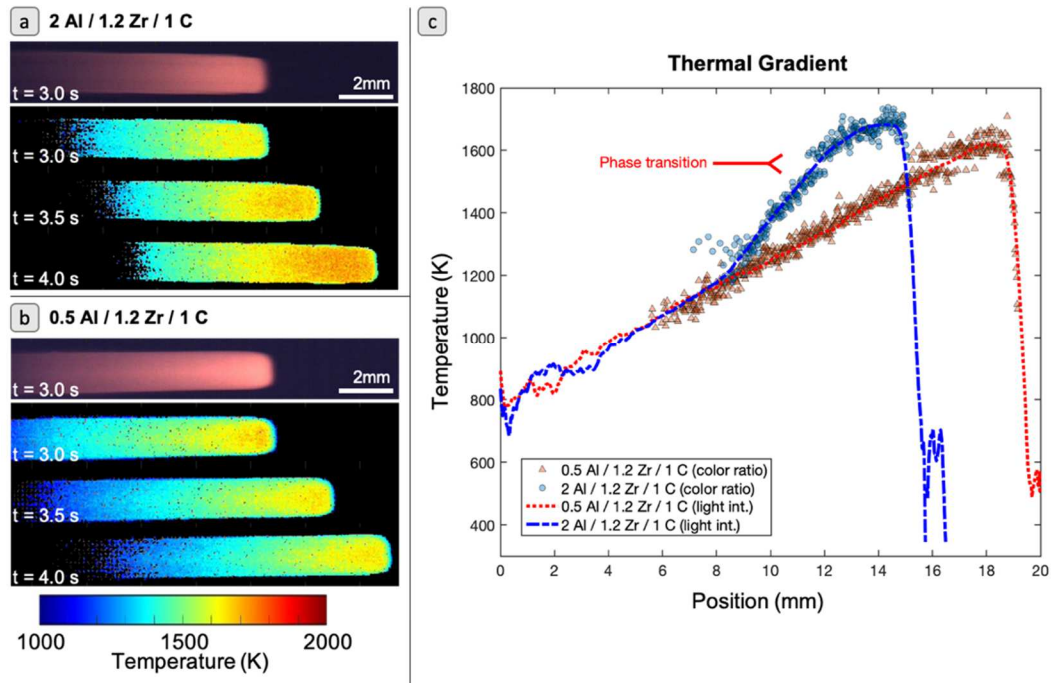
#### *6.4.2 Temperature profiles*

Conversion of the pixel intensities to temperature were performed using a home-built MATLAB routine that has been discussed in detail elsewhere.<sup>73</sup> The color ratio pyrometry code provides temperature information at each pixel location that is above the black level and not saturated. Figure 6-2 shows a typical thermal image via color ratio pyrometry for the propagation of the samples for a single frame without any spatial averaging. Light emission-based temperature measurements are also plotted in Figure 6-2 and show a nearly perfect overlap with the color ratio temperature measurements. The accuracy in the light emission temperature measurements suggests that this correlation can be used to extend the dynamic range of our temperature measurements to estimate the rise time, decay time, and thermal diffusivity of the leading edge as will be described in the next section.

Example thermal gradients (Figure 6-2) for the tested samples demonstrate two clearly different profiles. Lower Al content is associated with both wider leading and trailing edges of the thermal profile. The peak temperature of the samples is also sustained over a shorter distance for the sample with lower Al content, which is to be expected considering that the thermal conductivity of the samples should decrease with lower Al content. A similar relationship between the width of peak temperature sustainment and the Al content has also been previously reported by Hu et al.<sup>193</sup> The inverse relationship between the width of the trailing edge and Al content is counter to



the expectation since the Al content should increase the overall thermal conductivity of the samples, though if the final product is largely a Zr/Al alloy in both samples, then these results are attributable to varied thermal properties between different alloys. A more detailed discussion is provided in the following section on thermal diffusivity. Interestingly, the peak temperature of the reaction seems to be independent of the Al content since both composites show a peak temperature  $\sim 1600$ - $1700$  K. This is counter to what would be expected as the reaction temperature was speculated to be adversely affected by side reactions of Al and Zr with increasing Al content. However, the reduction in Al content would limit the prevalence of the Zr/Al melt and the dissolution of C into the system would be inhibited, thus decreasing the degree of completion of the ZrC formation reaction, reducing the total energy release, and lowering the peak temperature.



**Figure 6-2:** (a,b) Sample color image and montage of temperature data collected via color ratio pyrometry. (c) Example of temperature vs. position data for both samples

tested with regions of suspected phase transitions. Propagation direction is from left to right. Profiles shown are chosen for optimal comparison and not for the same timestamp. Dotted lines in the graph represent the light emission-based temperature measurements that are used to extend the dynamic range of the instrument.

One interesting feature of these temperature profiles is the apparent step change in temperature. This is most evident in both samples between  $\sim 1500$ - $1600$  K. This abrupt change in temperature is attributable to an endothermic phase transition, implying that there also has to be a significant increase in entropy to make the free energy thermodynamically favorable. The Al-Zr phase diagram is very rich where many equilibrium or non-equilibrium phase transitions could occur. A possible source of this endothermic phase transition is at  $\sim 1550$  K from a mixture of  $\text{Zr}_5\text{Al}_4$  ( $\Delta H_f = -0.4$  eV/atom) and  $\text{Zr}_2\text{Al}_3$  ( $\Delta H_f = -0.5$  eV/atom) to a mixture of  $\text{ZrAl}$  ( $\Delta H_f = -0.27$  eV/atom) and  $\text{Zr}_2\text{Al}_3$ .<sup>203</sup>

There was also a phenomenon observed in both samples where the light intensity would cyclically spike during sample propagation, though this was most prominent in the 2 Al / 1.2 Zr / 1 C case. A plot of light intensity vs. time for the 2 Al / 1.2 Zr / 1 C (Figure 6-7) exhibits periodic oscillations occurring at a frequency of 1 Hz (1 cycle/s,  $\sim 0.36$  cm/cycle) with temperature oscillations of  $\sim 50$ - $100$  K. One possible cause of the oscillations is that the sample is propagating in a “relay-race mode” where there are local hesitations in the reaction front because of thermal resistances between distinctly separated reaction regions.<sup>177,185,186,202</sup> However, it is unlikely that the sample is propagating in relay-race mode since the oscillations are periodic and that the length scale of the reaction front ( $\sim 1$ - $2$  mm, Figure 6-2c) is roughly 2-3 orders of magnitude larger than that of the composite powders (Figure 6-5).<sup>185-</sup>

<sup>187,202,204-206</sup> Videos recoded of the flame front with enhanced magnification (Figure

6-8) also show no evidence of resolved flame corrugation, casting further doubt on the notion of relay-race mode propagation. All of these observations, in addition to the linearity of the velocity profiles, suggest that this SHS system is within the quasi-homogenous propagation regime.<sup>202</sup> Such auto-oscillations have been reported in a variety of SHS systems and are widely agreed to originate from unsteady heat transfer in the reacting medium, whether it be driven by limitations in heat and mass transfer rates or multi-step reactions in SHS.<sup>178,207–211</sup> Unlike many of the studies that seek to numerically evaluate the oscillations in these materials, we do not see any hesitations in the propagation velocity. Overall, the origins of these oscillations are complex and not entirely understood.<sup>212</sup> The observation is noted here for thoroughness, but is not considered further.

#### 6.4.3 Extraction of thermophysical properties

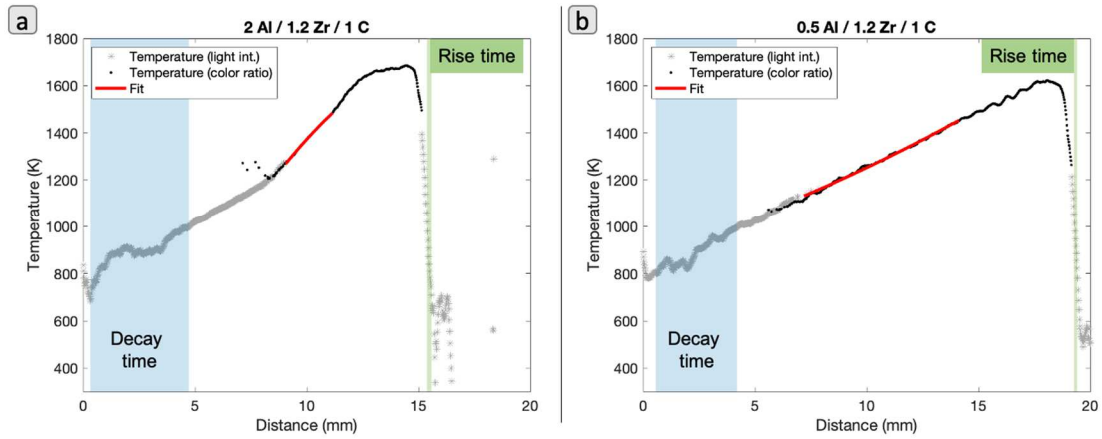
Thermophysical properties were extracted using a mixture of methods with data from the color ratio pyrometry and light emission-based temperature measurements. The resulting thermal diffusivity data has been summarized in Table 6-2. Example plots of the temperature data in the different regions, polynomial fits for the trailing edge estimation, and regions used for the rise and decay regions can be seen in Figure 6-3.

As discussed in the experimental section, the light emission temperature measurements were used as inputs for the Boddington equation. At temperatures below 933 K, it can be reasonably assumed that there is negligible chemistry and therefore the left-hand side of the Boddington equation can be set to 0. Built-in MATLAB routines to identify the rise and decay times of the SHS reaction, temperature measurements, and combustion velocity were used to estimate thermal diffusivity. For

the leading edge of the samples, the thermal diffusivity was nominally  $\sim 2 \times 10^{-6} \text{ m}^2/\text{s}$  between both samples. Although these thermal diffusivity values are relatively low, the numerous interfaces between the  $\sim \mu\text{m}$  sized particles and different materials within the particles would lower the thermal conductivity of the sample and therefore these values are believed to be reasonable.

On the trailing edge of the sample, the thermal diffusivity increased by  $\sim 30\text{-}40\times$  for both samples. This increase in thermal diffusivity on the trailing edge is the primary goal of Draper's RAIL3D program to create printable, reactive materials that have a higher conductivity after reaction.<sup>188,190</sup> This diffusivity enhancement suggests that the reaction is mostly complete before the temperature begins decreasing. The high temperatures reached during the reaction also results in particle melting and sintering which promotes conductivity/diffusivity. This result is also consistent with those reported in Arlington et al.<sup>188</sup> Interestingly, samples with higher Al content had a lower thermal diffusivity, which may seem counterintuitive since the final products are ideally Al and ZrC (Eq. 6-2 – 6-6) and Al is a very good thermal conductor. This behavior is likely traceable to the role of Al as a diluent in the reaction, thus larger amounts of Al could detract from the energy output of the reaction, reduce the ability for Al to reflow heat, and lead to incomplete reactions which has products/morphologies with lower thermal diffusivity.<sup>188,193</sup> The depression in the measured thermal diffusivity compared to the values of the pure components can be largely attributed to the introduction of a polymer binder for the printing process, interfacial resistances between the powder grains, and porosity in the materials. For

reference, the thermal diffusivities of Al, Zr, and C are roughly  $9.7 \times 10^{-5} \text{ m}^2/\text{s}$ ,  $1.2 \times 10^{-5}$ , and  $2.2 \times 10^{-4} \text{ m}^2/\text{s}$ , respectively.



**Figure 6-3:** Temperature profile data for (a) 2 Al / 1.2 Zr / 1 C and (b) 0.5 Al / 1.2 Zr / 1 C printed samples. Solid line in each plot shows the goodness of fit for the exponential curve used to estimate thermal diffusivity on trailing edge. Shaded regions highlight the regions used for rise and decay time where temperature is <933 K.

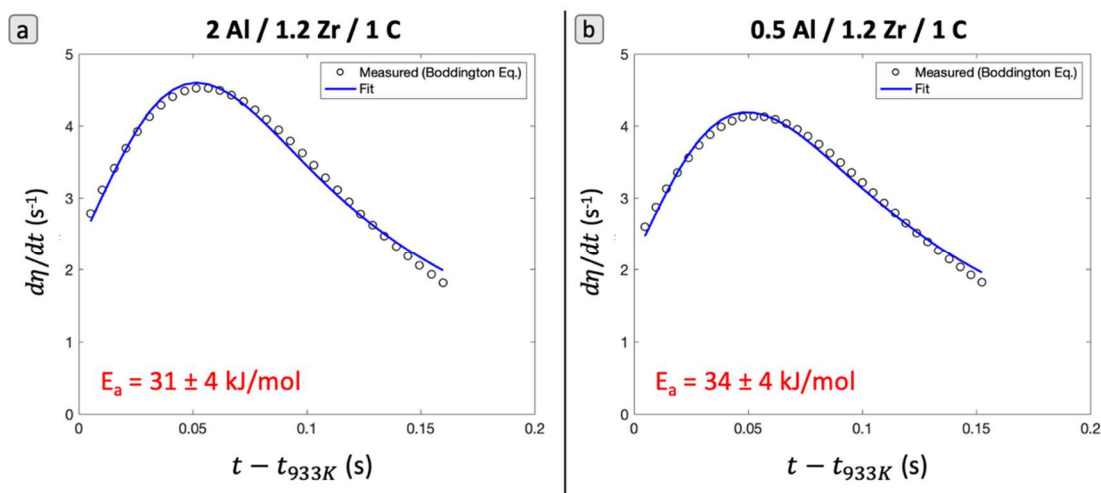
**Table 6-2:** Calculated thermal diffusivities for tested SHS samples as estimated using single line temperatures and polynomial fits. Estimated kinetic parameters for tested SHS samples estimated using nonlinear regression fit and model described by Equation 6-8.

Sample	Avg. Thermal Diffusivity ( $\alpha$ ), $\text{m}^2/\text{s}$		Kinetic Parameters		
	Leading	Trailing	A ( $\text{s}^{-1}$ )	n	E <sub>a</sub> (kJ/mol)
2 Al / 1.2 Zr / 1 C	$2.2 \pm 0.5 \times 10^{-6}$	$7.2 \pm 2.2 \times 10^{-5}$	$122 \pm 60$	$2.3 \pm 0.8$	$31 \pm 4$
0.5 Al / 1.2 Zr / 1 C	$2.6 \pm 0.1 \times 10^{-6}$	$1.1 \pm 0.1 \times 10^{-4}$	$184 \pm 79$	$2.6 \pm 0.5$	$34 \pm 4$

#### 6.4.4 Kinetic parameters

Arrhenius parameters and reaction order in Equation 6-8 were determined using a nonlinear regression fit of Equation 6-8 to the experimental determined  $d\eta/dt$ . Activation energies reported have been calculated for temperatures on the leading edge of the thermal profile ( $T > 933 \text{ K}$ ). A summary of the kinetic parameters use as inputs for Equation 6-8 can be seen in Table 6-2.

Plots of  $d\eta/dt$  versus time and corresponding Arrhenius model fits of the two mixtures are presented in Figure 6-4. The Arrhenius model fits are in good agreement with the measured data for  $d\eta/dt$ . Pre-exponential factors (A) were  $\sim 122 \pm 60 \text{ s}^{-1}$  and  $\sim 184 \pm 79 \text{ s}^{-1}$  for the 2 Al / 1.2 Zr / 1 C and 0.5 Al / 1.2 Zr / 1 C curve fits, respectively. The reaction order(s) (n) in the curve fits were nominally between 2.3-2.6, which is representative of a complex, multistep reaction mechanism proposed for the system.<sup>180,191-193</sup> Activation energies for the 2 Al / 1.2 Zr / 1 C and 0.5 Al / 1.2 Zr / 1 C composites were estimated to be  $\sim 31 \pm 4 \text{ kJ/mol}$  and  $\sim 34 \pm 4 \text{ kJ/mol}$ , respectively. These activation energies are lower than values that were previously reported for similar Al-Zr-C composites when estimated using differential thermal analysis (99-186 kJ/mol).<sup>179,193</sup> It is important to note, however, that Boddington et al. also reported lower activation energies<sup>40</sup> which they attributed to discrepancies in experimental conditions.<sup>195</sup> In particular, they suggested that kinetic analysis under ignition conditions will yield a different apparent activation energy than that obtained under steady reaction propagation as was done in in this work. This is not particularly surprising as they are very different processes and similar behavior has been reported for gas phase combustion. Therefore, the activation energy measured here represents the thermal sensitivity of the mass burning rate, which is a convolution of the thermal sensitivity to heat/mass transfer and the chemistry.



**Figure 6-4:** Plots of  $d\eta/dt$  vs. time and corresponding Arrhenius model fit used to calculate the activation energy of (a) 2 Al / 1.2 Zr / 1 C and (b) 0.5 Al / 1.2 Zr / 1 C samples. Timestamps were normalized to time after the reaction front profile exceeded 933 K.

The slight change in the estimated activation energies for the different composites is likely not significant, although it has been previously reported that an increase in Al would promote the Al-Zr intermetallic reaction which has a lower activation energy than the ZrC formation reaction.<sup>193</sup> Again, these values are meant to be representative of the bulk material under experimental conditions which could be used as inputs to a thermal propagation model.

## 6.5 Conclusions

This work investigates the propagation and thermophysical properties of an additively manufactured, self-propagating high-temperature synthesis (SHS) Al-Zr-C composite. By employing high-speed color ratio pyrometry, the temperature profile of the SHS compound was measured in-operando for two samples with varied stoichiometry. High-speed videography and pyrometry analysis of the reaction event showed that propagation velocity of the samples was 0.3-0.5 cm/s and varied inversely

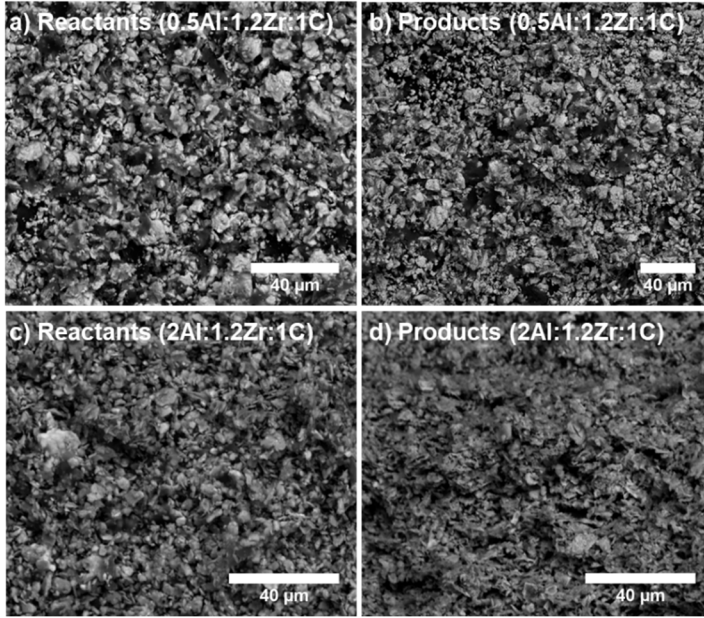
with Al content. These propagation events also demonstrated auto-oscillations in reactions associated with temperature rises of ~50-100 K at a frequency of 1 Hz. Using the temperature measurements, the thermal diffusivity of the samples was estimated using the thermal diffusion equation to be  $\sim 2 \times 10^{-6} \text{ m}^2/\text{s}$  on the leading edge of the reaction front and both samples exhibited a >30x increase in thermal diffusivity on the trailing edge. The increase in thermal diffusivity exhibited the desired behavior of have a material undergo a reaction which results in a conductivity enhancement over the starting material. Activation energy of the two samples was estimated to be ~30-35 kJ/mol. This work ultimately demonstrates a new, accessible measurement methodology that could be used to estimate thermophysical changes in materials for generalized modeling purposes.

## ***6.6 Acknowledgements***

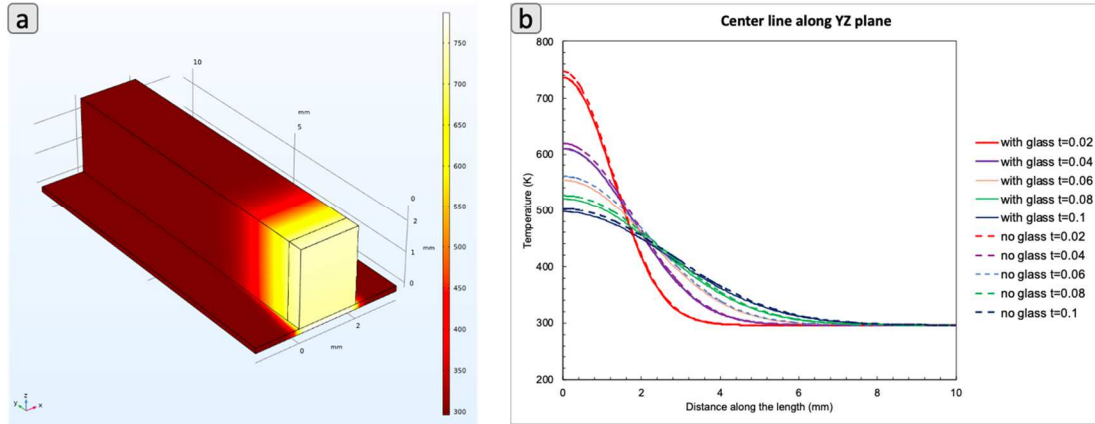
The authors would like to acknowledge the Charles Stark Draper Laboratory (CSDL) for providing materials for this study. This work was funded by the Army Research Office. Reproduced from D.J. Kline, M.C. Rehwoldt, J.B. DeLisio, S.C. Barron, H. Wang, Z. Alibay, J.C. Rodriguez, G.M. Fritz, M.R. Zachariah, In-operando thermophysical properties and kinetics measurements of Al-Zr-C composites, *Under review*.



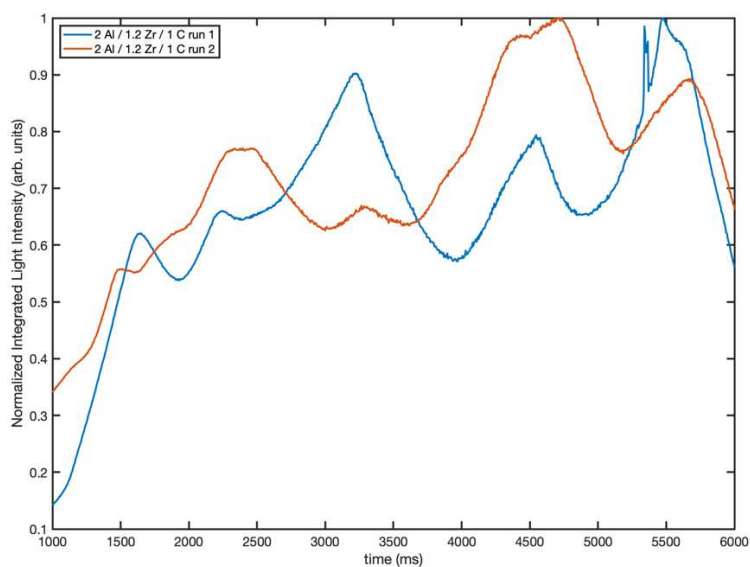
## Supplemental Information



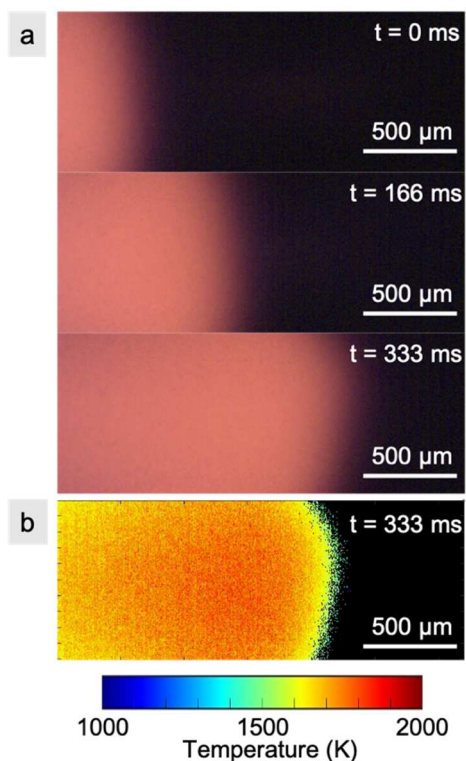
**Figure 6-5:** SEM visualization of reactant and product morphology of (a-b) 0.5Al:1.2Zr:1C and (c-d) 2Al:1.2Zr:1C stoichiometries.



**Figure 6-6:** (a) Depiction of COMSOL simulation to check influence of glass slide on temperature profile of the system. Figure includes glass slide. Initial temperature of hot region at  $t=0$  is 1700 K. (b) Temperature profile in YZ plane for different times in simulations with and without glass slide.



**Figure 6-7:** Plot of normalized light intensity vs. time for the 2 Al / 1.2 Zr / 1 C composite to demonstrate the periodic oscillations in brightness during reaction propagation.



**Figure 6-8:** (a) High-speed microscope images of propagation for the 0.5 Al / 1.2 Zr / 1 C samples and (b) corresponding temperature map as measured by color pyrometry. Images captured using a Phantom VEO710L high speed camera coupled to an InfinityOptics K2 DistaMax Lens with a resolution of  $\sim 1.7 \mu\text{m}/\text{px}$ .

## Chapter 7: Why does adding a poor thermal conductor increase propagation rate in solid propellants?

### **7.1 Summary**

Solid propellant additives have a long history of modulating burning rate by introducing materials with high thermal diffusivities to better concentrate and transfer heat to nearby areas. However, recent studies have demonstrated a counterintuitive result in that additives with thermally insulating properties – notably  $\text{SiO}_2$  particles – can also enhance the propagation rate in solid propellants. In this work, high-speed microscopy and thermometry were performed on 3D printed solid propellant films containing both thermally conducting (graphite) and insulating ( $\text{SiO}_2$ ) to investigate the role of these additives on film propagation rate. It was found that addition of  $\text{SiO}_2$  particles increased the effective surface area of the reaction front through inhomogeneous heat transfer in the films, and that such corrugation of the reaction front area on the micron scale manifests itself as a global increase in the propagation rate on the macro scale. Graphite additive was observed to have a substantially lower burning surface area and propagation rate, suggesting that the effect of reaction front surface area is larger than the effect of thermal diffusivity for low-weight percent additives in solid propellants.

### **7.2 Introduction**

Energetic materials – the overarching class of pyrotechnics, propellants, and explosives – have received a renewed interest in the scientific community with investigations into new formulations,<sup>6,55,73,74,91,213</sup> physical architectures,<sup>86,214,215</sup> and

expanding applications to the civilian and military communities. Considering new advances in nanomaterial production and additive manufacturing, solid rocket propellants have experienced a particular rise in research aimed at determining the underlying thermochemical mechanisms that govern their combustion performance and capitalize on their customizability and ease of production.<sup>86,87,181,215</sup> Among the most important parameters of solid propellants driven by the chemical formulation are the combustion temperature and the propellant burn rate.<sup>5</sup> Temperature of combustion dictates the theoretical maximum force that can be achieved in the expanding gases given a constrained velocity while propagation rate determines the time over which this force can be applied. Considering that only certain materials are traditionally employed as fuels (Al) and binders (HTPB) in solid propellants, modifying the propagation rate is a primary route by which to modulate the combustion performance. In the gas phase, the steady-state propagation velocity ( $v$ ) theoretically scales with the thermal diffusivity ( $\alpha$ ) of the combusting material and the chemical reaction rate ( $\dot{\omega}$ ) as shown in Equation 7-1.

$$v \propto \sqrt{\alpha \cdot \dot{\omega}} \quad (7 - 1)$$

One can reasonably conjecture that similar controlling processes might occur in the condensed phase. Modulation of combustion performance for solid propellants has a long history of capitalizing on embedding materials with a high thermal diffusivity. Space-race age reports by Caveny & Glick<sup>94</sup> demonstrate the principles upon which the addition of a wire with a high thermal diffusivity can dramatically increase the propagation rate in a solid propellant by enhancing heat conduction to different areas of the propellant. These “wired propellants” have since been widely used

and the dominating characteristics of their combustion performance is a distinct cone shape in the burning surface structure that is thought to increase the burning surface area, and thus, the propagation rate.<sup>176,216,217</sup> More recent experiments by Isert et. al capture the same cone structure and enhanced burning rate.<sup>176</sup>

Recent results by the Zachariah Group<sup>95</sup> and others<sup>218</sup>, however, demonstrate a uniquely different phenomena with the addition of thermally insulating particles that are embedded during material synthesis. Wang et al.<sup>95</sup> incorporated mesoporous silicon dioxide – a material commonly used for thermal insulation – into reactive nanoaluminum polyvinylidene fluoride (Al/PVDF) films prepared by electrospraying and noticed a ~3x increase in burn rate at 2.5 weight-% SiO<sub>2</sub>. To explain this, the authors suggested that the particles generated hot spots and effectively increased the number of ignition sites in the material, however direct evidence was not presented in the study.<sup>95</sup> Wang et. al also show evidence of meso-SiO<sub>2</sub> interaction with the fluorine compounds in the reactive system with TGA/DSC and MS, however, experiments with micron-SiO<sub>2</sub> at the same mass loading had a reduced overall decomposition likely attributable to the lower surface area of the micro-SiO<sub>2</sub>.<sup>95</sup> Thermogravimetric analysis by Shioya et. al showed no catalytic interaction of SiO<sub>2</sub> in their Al/AP/HTPB system, but also reported an increased overall burn rate with the addition of low weight-% SiO<sub>2</sub>.<sup>218</sup>

To determine why previous research has demonstrated an enhanced propagation rate in 3D printed solid propellants when doped with poor thermal conductors, two scenarios were evaluated: (1) addition of poor thermal conducting particles (SiO<sub>2</sub>) and (2) addition of a good thermal conductor (Graphite/Carbon). Films

were prepared by dissolving 300 mg polyvinylidene fluoride (PVDF, average molecular weight ~534,000) in 5 mL of dimethylformamide (DMF, 99.8%) and subsequently adding 300 mg aluminum nanoparticles (Novacentrix, ~85nm, ~81% active weight) for an equivalence ratio ( $\Phi$ ) of 2.87. The mixtures were then each sonicated for a half hour, after which 15.4mg of silicon dioxide micron particles ( $\text{SiO}_2$  1-5 $\mu\text{m}$ , Sigma Aldrich) and 15.4mg of graphite flakes (Alfa Aesar) were added to the two mixtures, respectively, for an average loading of 2.5% by weight. The two mixtures were then sonicated for another half hour and magnetically stirred for 24 hours. As prepared inks were then printed with a Hyrel 30M 3D onto microscope coverslips (VWR 0.17mm thickness, 22 mm square) over an 80C preheated plate in an 8cm-square pattern at a flowrate of ~0.3mL/min and speed of ~22 cm/min.

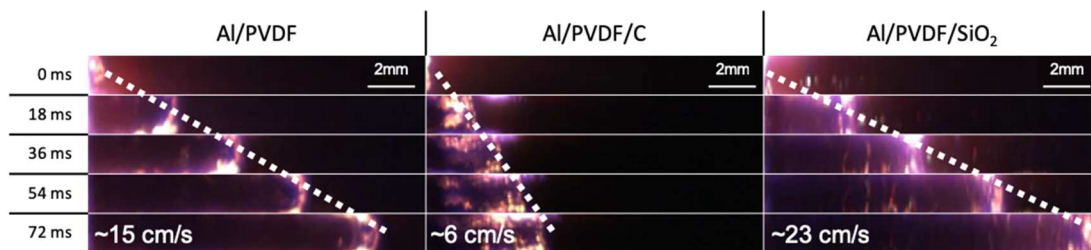
Scanning electron microscopy images of the doped Al/PVDF samples are presented in Figure 7-5. The Al/PVDF/C film on the left shows a relatively even distribution of the micron-sized graphite particles embedded into the printed film's cross section. The Al/PVDF/ $\text{SiO}_2$  film cross-section shows large agglomerations of particles and a much more cavernous cross section with  $\text{SiO}_2$  particles (1-5  $\mu\text{m}$ ) distributed within the frame. It is important to note that porosity of the films does alter the propagation rate in the materials, however this impact would likely be less evident for material that was firmly adhered to the microscope slides.<sup>86,87</sup>

### **7.3 Results**

#### **7.3.1 Macroscale combustion tests**

Combustion of the samples was then visualized using a Vision Research Phantom VEO710L camera at 5,000 frames/second at both the macro- and micro-scale.

In both sets of tests, samples printed on the coverslips were mounted vertically on a 3-axis translational stage (Newport), brought into focus, and ignited using a resistively-heated nichrome wire. For the macro tests shown in in Figure 7-1, the burn time was the time it took from ignition of the material to the completed propagation across the 22 mm slide. Quite clearly, it can be seen that, when compared to the nascent Al/PVDF base case, addition of carbon (a good thermal conductor) slows the reaction front, while addition of silica (a poor thermal conductor) speeds up the reaction front. This is in direct contradiction to that predicted by Eq. 7-1. We now turn to the microscopic imaging.



**Figure 7-1:** Macro-scale burn rate images of a printed film on a microscope slide at 5,000 frames/s. Burn time measured from ignition to the time that the reaction front reaches the end of the microscope slide. Note: Propagation is from left-to-right.

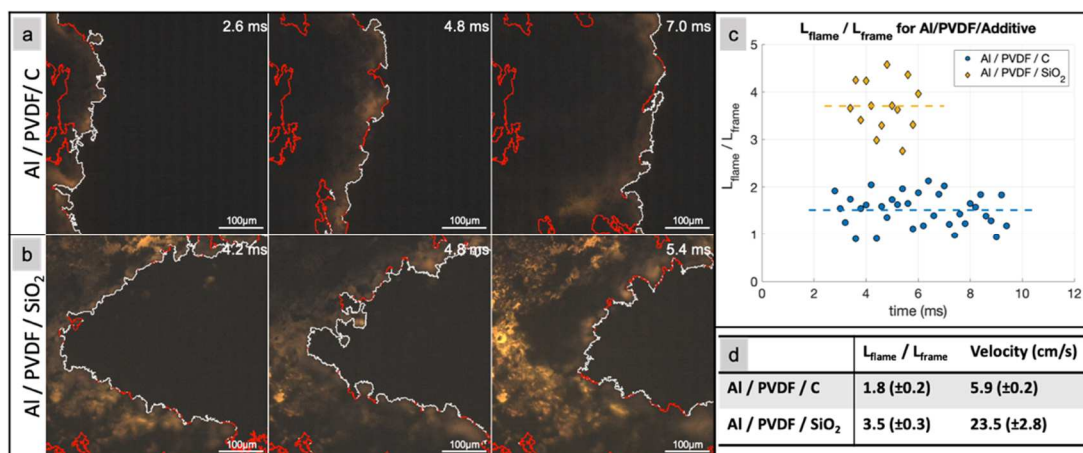
### 7.3.2 Microscale combustion tests

Evaluation of the combustion characteristics at the microscale used a house-built high-speed microscope assembly in which the ignition event is observed using light passed through a 40x microscope objective (Nikon) which is transmitted through a beam splitter, through a camera lens (Nikon 105 mm Macro) focused at infinity, and onto the camera sensor.<sup>219</sup> With the microscope objective, the pixel resolution of the camera assembly is  $\sim 1\mu\text{m}/\text{pixel}$  and allows us to resolve an approximate the reaction front length and initial topography of the film (see Figure 7-6).

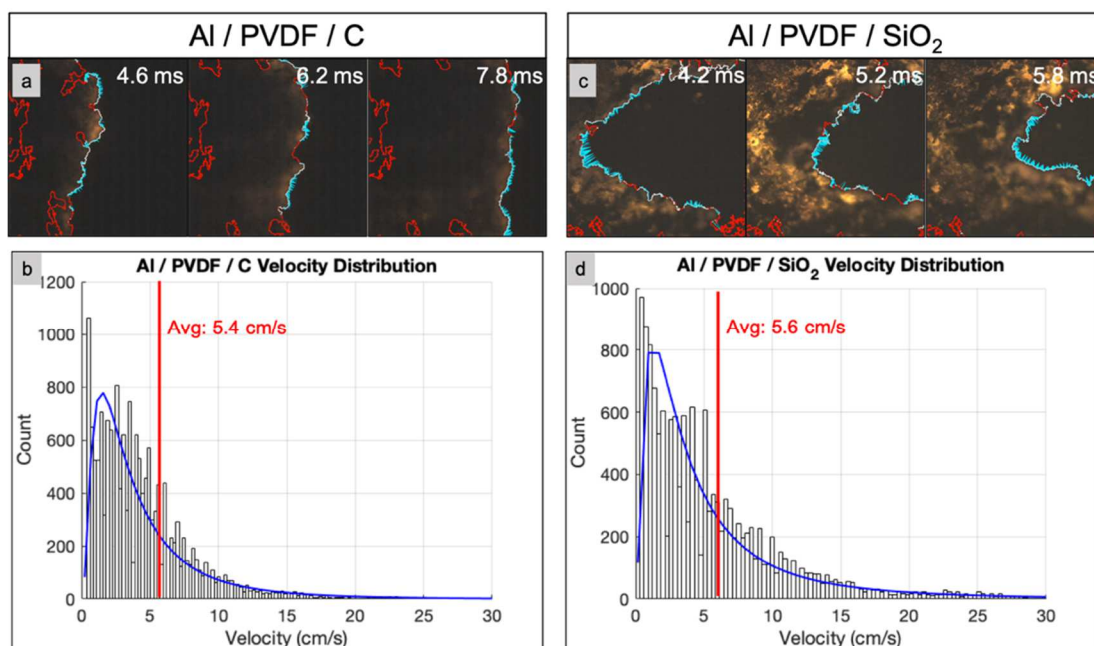
Microscale videos were post processed to determine an estimated reaction front length and velocity distribution. Images are imported using a custom MATLAB routine which provides raw pixel intensities for each image and the images are subsequently binarized using MATLAB's Image Processing Toolbox.<sup>167</sup> The length of the reaction front only considers those points that had moved within the previous two frames of the recorded video as identified by the binarization routine.

Figure 7-2 demonstrates the effect of C and SiO<sub>2</sub> additives on the effective reaction front length of the propagating Al/PVDF films. The Al/PVDF/C additive has a relatively linear reaction front that has larger corrugated areas spanning the entire length of the window (Figure 7-2a). However, the Al/PVDF/SiO<sub>2</sub> front has many more small corrugations spread out over the entire length which effectively increases the surface area of the burning front (Figure 7-2b). Using these results, the normalized reaction front length ( $L_{\text{flame}}/L_{\text{frame}}$ ) was plotted in Figure 7-3c, which clearly quantifies that both composites have a steady reaction front length for the few milliseconds that the film is within view, and that the length of the Al/PVDF/SiO<sub>2</sub> front is ~2x longer than that in the Al/PVDF/C case. When averaging the reaction front length over three separate tests, this 2-fold increase in reaction front length remains consistent (Figure 7-2d). Also from Figure 7-2d, the global-average velocity of the macro-scale burn tests appears to increase in accordance with flame length – a reasonable expectation since burn rate in solid propellants is directly related to the burning surface area.<sup>5</sup>





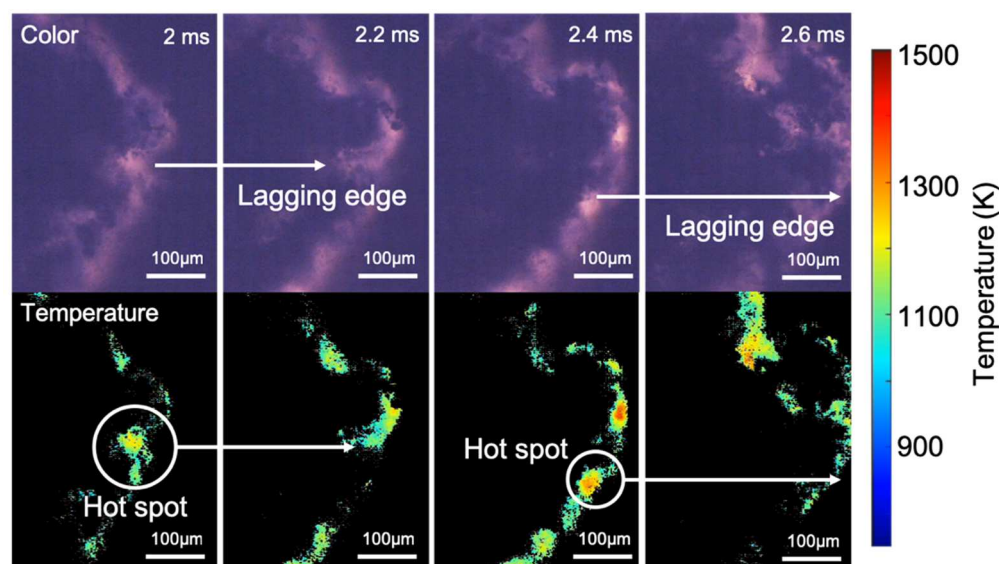
**Figure 7-2:** High-speed microscopy images for (a) Al/PVDF/C and (b) Al/PVDF/SiO<sub>2</sub> with outlines of reaction front. Red lines represent regions that had not moved within 2 frames. (c) Plots of window-size normalized flame length as a function of time for Al/PVDF/C and Al/PVDF/SiO<sub>2</sub>. (d) Average flame length and velocity for additive-doped films. Note: Propagation is from left-to-right.



**Figure 7-3:** (a) Al/PVDF/C and (c) Al/PVDF/SiO<sub>2</sub> imaged through high-speed microscopy with area outline (red/white) and velocity vectors (blue). Histogram of measured velocities for (b) Al/PVDF/C and (d) Al/PVDF/SiO<sub>2</sub>.

To further investigate these observations, the impact of the local velocity on the global average velocity observed was considered. Local velocities of the points along the reaction front were determined by using a closest-point determination between the active reaction front in one frame to the next frame and dividing by the framerate (5,000

fps). Sample images of the Al/PVDF composite films along with their reaction front areas and estimated velocity vectors are shown in Figure 7-3a/c, where a relatively constant velocity vector length along the Al/PVDF/C front and a wider distribution over the length of the Al/PVDF/SiO<sub>2</sub> front can be seen. Histograms of the Al/PVDF/C and Al/PVDF/SiO<sub>2</sub> calculated velocity vectors summed over the length of each video are presented in Figure 7-3b/d and, while the distribution of velocities is wider in the Al/PVDF/SiO<sub>2</sub> case, the average local velocities are both  $\sim 5.5$  cm/s. Considering local velocities are similar for both cases tested, the observed effect on global burn rate can be attributed to the corrugation in the reaction front rather than any specific enhanced chemistry that drives reactions faster.



**Figure 7-4:** (Top) High-speed microscopy images and (bottom) corresponding temperature maps obtained by color camera pyrometry.

Temperature maps of the high-speed microscopy images were used to evaluate the plausibility of inhomogeneous heat transfer on the reaction front for both additives. Briefly, temperature measurements can be extracted using the color video camera using channel intensity ratios for the different colors of the filter array placed in front of the

camera sensor, assuming graybody behavior of the burning material, and integrating the expected light intensity for each color channel for the entire light spectrum over which the camera is sensitive. More details on the methodology can be found in prior work.<sup>73</sup> Careful observation of the reaction fronts in the original color images show, for the Al/PVDF/SiO<sub>2</sub> samples, the emergence of distinct bright regions which lead to lagging edges in the following frames (see Figure 7-4). Pyrometry measurements made on these images show the evolution of hot spots with localized temperatures 300-400 K higher than the rest of the reaction front where the silica particles were observed in the “before image.” It is therefore believed that the lagging edge in the reaction front was induced by the additive particles which thereby increased the surface area of the burning front.

#### **7.4 Discussion**

As previously discussed, the burn rate of a solid propellant is estimated to scale directly with the burning surface area ( $A$ ), however it is also proposed to scale with the square root of thermal diffusivity ( $\alpha$ ) and the chemical reaction rate ( $\dot{\omega}$ ). When only considering the effect of thermal diffusivity on the propagation velocity, it would be expected that the carbon additive ( $\alpha \sim 200 \text{ mm}^2/\text{s}$ )<sup>220</sup> would burn much faster than the SiO<sub>2</sub>-doped composite ( $\alpha \sim 0.8 \text{ mm}^2/\text{s}$ )<sup>221</sup> because of faster heat feedback in the latter, but the exact opposite behavior was observed. Mass-average estimations of thermal diffusivity presented in Table 7-1 show that, on a bulk scale, the low weight-% additives change  $\alpha$  of the Al/PVDF  $\Phi=2.87$  ( $\alpha \sim 48 \text{ mm}^2/\text{s}$ ) less than 10%, suggesting that the observed burn rate modulation is dominated on the microscale where differences in local  $\alpha$  change by orders of magnitude.<sup>222</sup> Thus, the observed changed

in burn rate is likely because, at low mass loadings, the  $\text{SiO}_2$  additive is stochastically modulating the heat transfer, resulting in an in-homogeneous thermal profile. This in turn leads to corrugation of the reaction front, thus increasing burning surface area and the mass burning rate which, at the macroscale, is observed as an increase in the global reaction front velocity. Carbon, in contrast with its higher thermal conductivity, more evenly distributes the released energy along the burning surface so that all regions have effectively the same local velocity, resulting in a flatter reaction front with lower surface area. This mechanism of enhanced propagation velocity likely breaks down, however, at higher mass loadings or in scenarios where a concentrated region of high thermal diffusivity enhances heat transfer on a bulk scale and creates a burning rate gradient that also effectively increases the burning surface area.

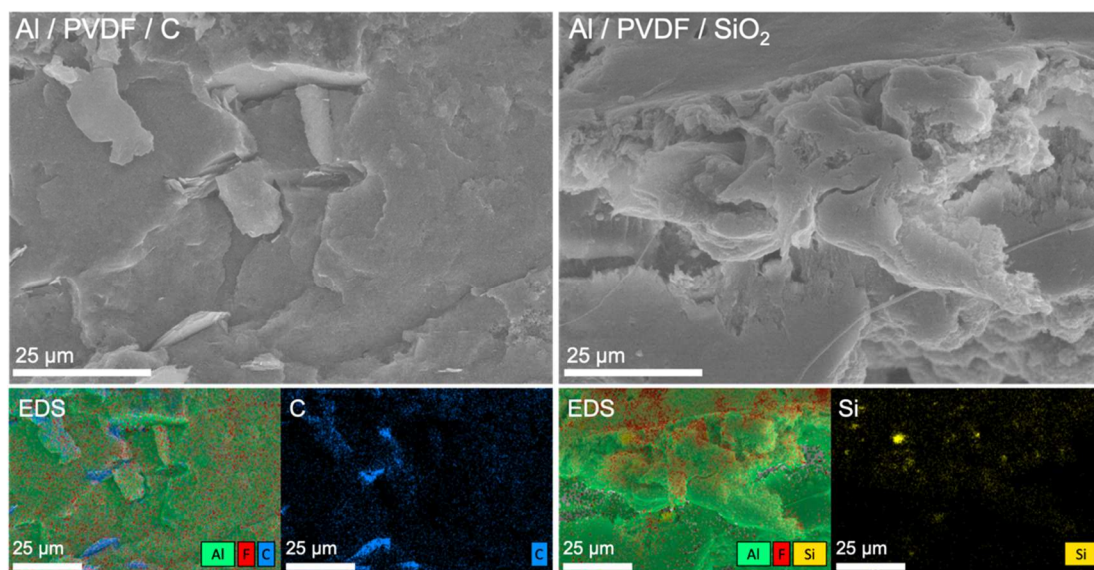
## ***7.5 Conclusions***

The counterintuitive observation that thermally-insulating additives increase burn rate in solid propellants can therefore be attributed to the role that burning surface area plays on the propagation rate. High-speed microscopy suggests that inhomogeneous heat transfer in the burning films leads to increased corrugation of the reaction front when the additives are homogeneously dispersed in the film at low mass-percentages. This observation therefore suggests that an alternative to large conductive wires being added to propellants in the manufacturing stage of rocket motors, engineers could instead introduce small masses of thermally-insulating particles to increase the effective burning surface area and burning rate.

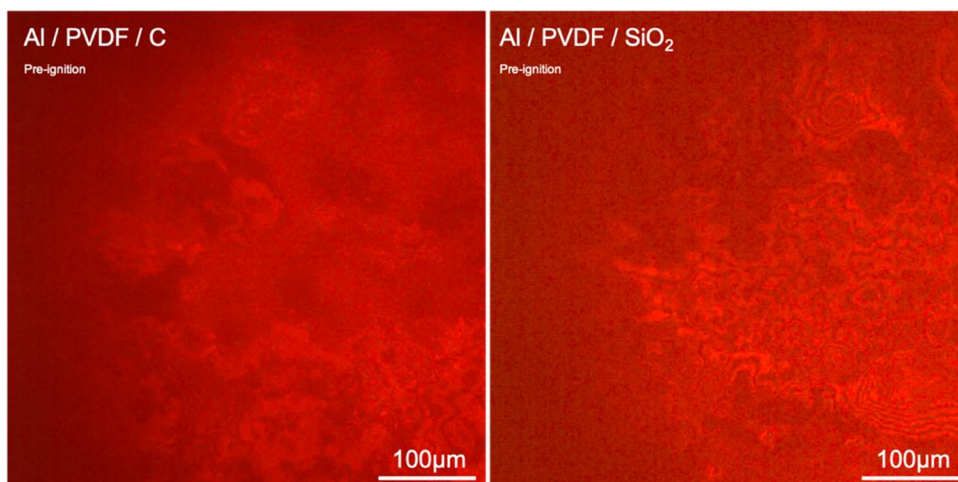
## ***7.6 Acknowledgements***

The authors gratefully acknowledge support from the Army Research Office and the Air Force Office of Scientific Research. Reproduced from D.J. Kline, M.C. Rehwoldt, H. Wang, N.E. Eckman, M.R. Zachariah, Why does adding a poor thermal conductor increase propagation rate in solid propellants?, Appl. Phys. Lett. 115 (2019) 114101. doi:10.1063/1.5113612 with permission from AIP Publishing.

### 7.7 Supplemental Information



**Figure 7-5:** Scanning electron microscopy images and elemental maps of typical cross-sections for Al/PVDF/C film (left) and Al/PVDF/SiO<sub>2</sub> (right) films showing distributions of large particle additives in each film.



**Figure 7-6:** Images of the Al/PVDF/C (left) and Al/PVDF/SiO<sub>2</sub> (right) films prior to ignition as observed against the microscope slide while focusing for the high-speed microscope.

**Table 7-1:** Estimated thermal diffusivity of different materials used in this study. Data presented for mixtures were calculated on a mass-basis.

Material	$\alpha$ (mm <sup>2</sup> /s)	Reference
Al	~ 97	NIST Webbook
PVDF	~ 0.1	223
SiO <sub>2</sub>	~ 0.8	221
C (graphite)	~ 165	220
Al/PVDF ( $\Phi=2.87$ )	~ 48	Calculated
Al/PVDF/2.5% SiO <sub>2</sub> ( $\Phi=2.87$ )	~ 47	Calculated
Al/PVDF/2.5% C ( $\Phi=2.87$ )	~ 51	Calculated

## Chapter 8: Numerically evaluating energetic composite flame propagation with thermally conductive, high aspect ratio fillers

### **8.1 Summary**

Thermally conductive, high aspect ratio additives offer a unique way to modulate heat transfer for flame propagation of energetic materials by creating “highways” along which released energy can transfer to unreacted regions. Flame propagation performance in dense energetic composites filled with randomly orientated additives of varied filler volume percentage (vol.%), aspect ratio (AR), and thermal conductivity was modeled using a 2D implicit finite element method. It was found that small additions ( $<20$  vol.%) of high thermal conductivity, high aspect ratio additives ( $AR > 25$ ) enhanced the total area burned by as much as  $\sim 25\%$  and the energy released by as much as  $\sim 10\%$ . While lower aspect ratio additives ( $AR < 25$ ) resulted in negligible enhancements due to the lack of thermally conductive pathways, large amounts of additives ( $>20$  vol.%) oftentimes led to quenched flame propagation by excessive leeching of energy from the reaction front. This study serves as a proof-of-concept for a method to enhance energy release by increasing heat transfer with additives in a method that could be readily incorporated into the energetic composite manufacturing process.

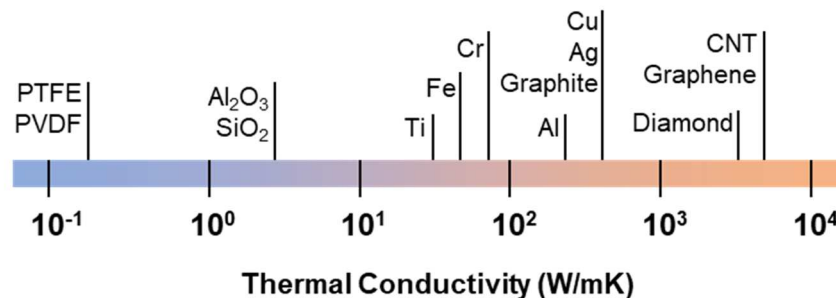
### **8.2 Introduction**

Advanced energetic and propellant systems are increasingly being viewed as good candidates for additive manufacturing. This is particularly so when considering the use of nanomaterials which make traditional casting difficult. These systems rely

on high volume loadings of a sub-micron metal fuel nanoparticles and an oxidizer either in the form of polymer binder or sub-micron source of condensed oxygen (e.g. metal oxides, ammonium perchlorate). For full utility, these materials must not only have high energy density, but must release its chemically stored energy in a rapid, stable manner and must be mechanically durable. One significant aspect of the operational characteristics of self-sustained combustion in these materials is heat feedback to maintain flame front propagation, which implies that thermal transport properties play an important role. These materials fall into a much broader class of polymer composites.

Characterizing the tunability of thermal and mechanical properties of multi-phase<sup>224</sup> polymer systems has been difficult to explore both experimentally<sup>225–228</sup> and theoretically<sup>229–231</sup> despite its importance in a variety of applications which include energy storage for electronics<sup>227,232</sup>, electrical shielding<sup>225,233</sup>, and heat exchange technologies.<sup>225,227,234</sup> Commonly used epoxies, acrylates, and fluoropolymers are poor conductors of heat and are generally treated as thermal insulators.<sup>235</sup> Thermally conductive fillers like carbides<sup>227</sup>, metals<sup>148,225</sup>, and carbon structures<sup>226,234,236,237</sup> with various morphologies (i.e. fibers, spheres, flakes) are oftentimes utilized to modulate the composite's effective thermal conductivity. General candidate materials span a wide range of thermal conductivities as can be seen in Figure 8-1. Improvements to thermal conductivity in composites has been primarily observed with the addition of thermally conductive fillers such as silver particles<sup>225</sup>, graphite<sup>237</sup>, carbon fibers<sup>236</sup>, and carbon nanotubes<sup>228,234</sup> (CNTs).





**Figure 8-1:** Spectrum of thermal conductivities typically used in polymer composites with tunable thermal properties.

Despite the widespread experimental and theoretical work studying the effects of thermally conductive materials on the effective thermal conductivity of two-phase polymer systems, little modeling work has investigated the extent to which the inclusion of such materials effects flame propagation performance in self-propagating high temperature synthesis (SHS)<sup>179,180,206</sup> and propellant systems.<sup>148,176,224,228</sup> Systems of primary interest are those which imbed condensed phase energetic constituents in the form of micron to nanoscale metal fuels, oxidizers, and/or other applicational additives of various morphologies within a polymer matrix binder.<sup>34,175,238–245</sup>

This paper explores the inclusion of materials with high thermal conductivity a means to augment the energy release rate in a dense composite.<sup>246</sup> In the absence of convective heat transfer, flame propagation behavior is dependent on how fast energy can be generated (chemistry) and how fast that released energy can be transferred to unreacted areas (thermal diffusivity). Thus, for a composite material with constituents having roughly the same order of magnitude of the specific thermal mass ( $\sim 10^6 \text{ Jm}^{-3}\text{K}^{-1}$ ),  $\rho c_p$ , the local flame behavior is explicitly a function of the local thermal conductivity.<sup>1</sup> The role of filler material parameters (e.g. volume percentage, material type, and geometric morphology) on composite properties have been explored with

experimental studies which reference effective composite thermal conductivity theories, such as the Lewis-Nielson, Maxwell, and Percolation models.<sup>225,226,229–231</sup> Experiments and Lewis-Nielson models have shown that the use of rods or fibers results in superior effective composite thermal conductivity over sphere-like morphologies at similar volume fractions since high aspect ratio additives tend to easily form a connective network along directions of interest.<sup>228,231,234,236</sup>

Here, a 2D implicit finite difference numerical model of the heat equation is used to analyze flame propagation behavior of energetic systems with low thermal conductivity (e.g. Al/PVDF) which incorporate thermally conductive fillers of varied volume percentages (Vol.%), aspect ratios (AR), and thermal conductivity. More specifically, this work focuses on a cross sectional area parallel to flame propagation in which rods of random azimuthal orientation lay in the plane. Results illuminate how thermal properties and materials distribution within the energetic composite correlates to flame front morphology, permeability of heat, rate of material consumption, and changes to overall energy release.

#### 8.2.1 Thermal transport and chemistry

Self-sustaining combustion of a dense reactive material involves heat transfer initiating rapid chemistry, which in turn drives more heat transfer.<sup>1</sup> The energy that is liberated during an exothermic reaction is converted to heat and observed as regions of temperature accumulation. Thermal gradients drive conductive energy transport from regions of high temperature to regions of low temperature through conductive pathways. For global flame propagation of a heterogenous reactive material to be self-sustaining, the combination of energy generated by chemistry and inward heat diffusion

in local regions must equal or exceed outward heat diffusion in order to sustain the global reaction front.<sup>1</sup> Rapid heat transfer from adjacent reacted materials ensures a rapid accumulation of energy to stimulate further chemical reactions. Often, the primary limitation to the observed propagation rate is that chemistry is so fast that the process is limited by the transport of heat to the unreacted region.

Under these conditions, the governing dynamics is described by the 2D Heat Equation (Equation 8-1) in which only Fourier conductive heat transfer flux,  $\vec{q}_{cond} = -k\vec{\nabla}T$ , between discrete material regimes is considered with the assumption that the thermal conductivity is constant along conduction boundaries. Additionally, external volumetric heat input,  $P$ , and internal chemical energy generation,  $Q$ , need to be considered. Handling of the chemical generation term will be discussed in a later section.

$$\rho C_p \frac{dT}{dt}|_{x,y} = \frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right)|_x + \frac{\partial}{\partial y} \left( k \frac{\partial T}{\partial y} \right)|_y + (P + Q)|_{x,y} \quad (8 - 1)$$

It is assumed that material density is sufficiently large for mass diffusivity to be negligible compared to thermal diffusivity ( $Le \ll 1$ ) so that mass diffusivity and convective behavior may be ignored.<sup>202,247</sup> The effective local thermal conductivity at the boundaries of discrete material regimes is determined by estimating thermal resistance (Equation 8-2).<sup>202,224,230,248</sup>

$$k_{eff} = \frac{2k_1k_2}{k_1+k_2}, \quad k_{p,q+\frac{1}{2}}^n = \frac{2k_{p,q+1}^n k_{p,q}^n}{k_{p,q+1}^n + k_{p,q}^n} \quad (8 - 2)$$

The specifics of discretization and numerical methods and solutions to the 2D Heat Equation are described in a supporting document in the published version of this article (see section 8.6 ).

The initial volumetric power input,  $P$ , (e.g. hot wire, laser, electrical) heats the material to the pre-defined ignition temperature,  $T_{ig}$ , and initiates volumetric chemical energy generation,  $Q$ . Since the intent is to focus on the role of heat transfer media, chemistry is simplified as the product of the global specific reaction enthalpy of the energetic material,  $\Delta h_{rxn}$ , with the global conversion rate of the of the reactant material,  $\frac{\partial \eta}{\partial t}$ , which is proportional to the local concentration of reactant species.<sup>204,249–</sup>

251

Several assumptions of chemistry are made to simplify the model while maintaining the integrity of analyzed trends:

- Chemical reaction time scale is much shorter than that of diffusion ( $t_{rxn} \ll t_{diff}$ ).<sup>204,205,252,253</sup>
- Chemical kinetics is linearized with the assumption of a sufficiently large activation energy for the onset of chemical energy output at  $T_{ig}$  (Heaviside Function).<sup>202,204,206,250</sup>
- The adiabatic flame temperature,  $T_{ad}$ , serves as a surrogate for both the reaction enthalpy and the termination point of chemical energy generation (Heaviside Function).

Transient chemical energy generation is thus represented in Equation 8-3 in an implicit binary manner as a function of experimentally and theoretically determined temperature values ( $T_{ig}, T_{ad}$ ) without the need for chemical conversion tracking.

$$Q = \Delta h_{rxn} \frac{\partial \eta}{\partial t} \sim \rho c_p \frac{T_{ad} - T_{ig}}{t_{rxn}} H(T - T_{ig}) H(T_{ad} - T) H(t_{rxn} - t_{ig}) \quad (8 - 3)$$

### 8.2.2 *Physical layout*

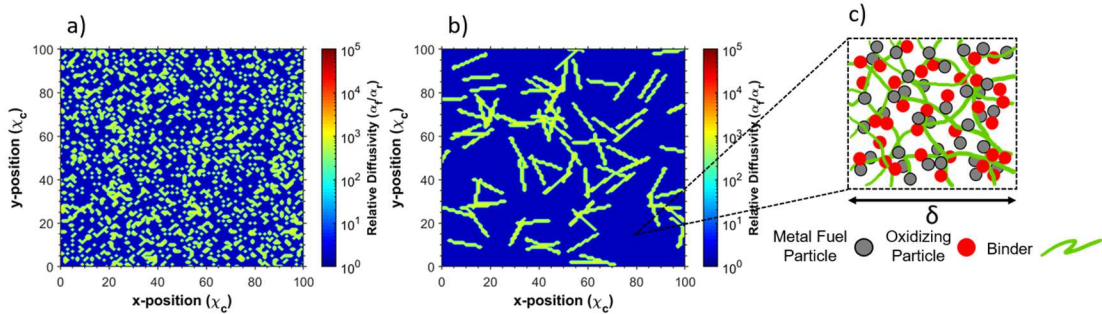
The physical platform of this model emulates that of the studies done by Varma<sup>202</sup>, Grinchuk<sup>254</sup>, and Tang<sup>187</sup> in terms of geometry and discrete composition layout (Figure 8-7). In accordance with a finite difference method, a constant volume<sup>187</sup> 2D mesh grid in a constant pressure environment is established with each grid point representing a temperature node at the center of a square area which has a characteristic width,  $\delta$ .<sup>255,256</sup> The grid represents an  $L_y \times L_x$  frame with  $\frac{L}{\delta}$  nodes along a single primary axis (grid resolution). Given the symmetry of the grid, one may choose to use the dimensionless length,  $\chi_c = \frac{x}{\delta}$ , where  $\delta < L$ .

The material properties of each node area represent a reactive material sample which is below or equal to its theoretical maximum density. This material properties may be varied to reflect the effects of composition morphology within the characteristic length. Each node area represents a composite of perfectly mixed implicit solid constituents which are no larger than the characteristic length. Applicable systems of this model must be constrained by the following assumptions:

- The scales of primary particle heterogeneity within the energetic composite is smaller than the characteristic length (Figure 8-2c).
- The thickness of the flame front is larger than the particle size and/or particle interspacing so that local flame propagation is treated as approximately quasi-homogenous.<sup>182,185,187,202,204,246</sup>

With these assumptions, specific geometric distribution of constituents within each square cell,  $\delta^2$ , can be ignored and one can simply use bulk quantities. One may

effectively vary the magnification of a system to different length scales while maintaining the relative resolution of the grid (at the potential cost of solution accuracy). Each temperature node consists of an effective bulk thermal conductivity,  $k$ , specific heat,  $c_p$ , and local density,  $\rho$ . A two-phase mixture<sup>224</sup> is implemented by designating each node to be either energetic material or inert filler. This allows for a single effective thermal diffusivity value,  $\alpha$ , to represent a single node of filler material,  $\alpha_f$ , or energetic material,  $\alpha_r$ . The filler material is a non-reactive material meant to simulate the physical existence of a material pore, longer range ( $>\delta$ ) heterogeneity, a conductive additive, or any other relatively non-reacting niche filler. A mapping of discrete thermal diffusivities/conductivities of each grid point is generated as a visual for material distribution, as shown in Figure 8-2.



**Figure 8-2:** Relative thermal diffusivity ( $\alpha_f/\alpha_r$ ) mapping with respect to volume fraction for (a) simple homogenous mixed, uniform distribution, and (b) distribution of rod of variable aspect ratios as well as an example of (c) the heterogeneity scaling of solid energetic constituents relative to  $\delta$ .

The placement of filler material is randomized by house-built routines which aim to match the desired volume fraction of evenly distributed filler for a given calculation. Figure 8-2b represents fillers with a non-spherical morphology in the form of rods whose centers of mass are uniformly distributed. The aspect ratio of the additive (AR) is monodisperse and varied between calculations. However, the minimum

geometries of the rods are constrained by the model to be no smaller than the characteristic length. As the aspect ratio of the sample approaches unity (AR=1), the distribution resembles the simple homogeneously mixed case shown in Figure 8-2a. As discussed above, a primary assumption is that the length scale of the energetic composite's components is much smaller than the resolution of the grid, therefore the energetic material appears to be perfectly mixed (Figure 8-2c).

### 8.2.3 Model analysis and parameters

The objective of this model is not to calculate specific numbers, but rather to flush out general trends for flame propagation behavior as a function of thermal properties, volume fraction of constituents, and the aspect ratio of incorporated rods. Although this model does not consider a specific system, many of the parameters for experimental conditions and thermophysical properties reflect those that would be seen in typical studies on fluoropolymer-bound energetic composites.

Modeling of flame behavior is assessed with room temperature initial conditions ( $T_a = 300$  K) using  $\frac{L}{\delta} = 100$  (e.g.  $\delta = 2 \mu m$ ,  $L = 200 \mu m$ ). Boundary conditions are set to allow for convective heat flux,  $\dot{q}_{conv} = h(T - T_a)$ , between the material and environment. The nonlinearity of radiative heat transfer disqualifies its contribution in this model which utilizes linear algebraic methods.

Thermal properties of interest for conductive heat transfer flame modeling consist of the thermal mass ( $\rho c_p$ ), thermal conductivity ( $k$ ), ignition temperature of the energetic material regimes ( $T_{ig}$ ), adiabatic flame temperature of energetic material chemistry ( $T_{ad}$ ), and characteristic time of reaction ( $\tau_{rxn}$ ). Given that the thermal mass

for constituents of potential interest (e.g. Al, Ti, Cu, C, PVDF, Viton) have the same order of magnitude, the thermal mass of each material type throughout this analysis remains invariant at a value of  $10^6 \text{ Jm}^{-3}\text{K}^{-1}$ . The quantitative value of thermal conductivity of the energetic material has a lower bound based on the binder. Neat polymer binders such as Viton and PVDF have thermal conductivities of  $\sim 0.2 \text{ Wm}^{-1}\text{K}^{-1}$  which increase with the addition of thermally conductive fuels.<sup>77,170,234,257</sup> For simplicity, the energetic material possesses a thermal conductivity,  $k_o$ , of  $0.4 \text{ Wm}^{-1}\text{K}^{-1}$  (double PVDF)<sup>170</sup> with the inert rods having varied thermal conductivities higher than the energetic material. The chosen ignition temperature and adiabatic flame temperature are based on Al/CuO thermite composites which have a measured ignition temperature of  $\sim 900 \text{ K}$ , and a theoretical adiabatic flame temperature of  $\sim 2800 \text{ K}$ .<sup>34,166</sup> The rough order of magnitude of the time of reaction is based on quasi-homogenous flame propagation<sup>258</sup> and laminar flame theory<sup>246</sup> with experimental observations of Al/CuO propellants which satisfy the assumption of fast chemistry.<sup>166,258</sup>

### **8.3 Results**

#### **8.3.1 Role of rod aspect ratio in combustion performance**

Flame propagation modeling was carried out over a  $500 \mu\text{s}$  time period. Energetic composites are incorporated with randomly oriented rods at varied volume percentages (vol.%=0%, 5%, 10%, 20%, 30%, 40%) and aspect ratios (AR=1, 5, 15, 25, 35, 45, 55). Modeling of each parameter set was repeated 5 times. Temperature mappings in Figure 8-9 illustrate overall flame morphology with hot spots designating zones undergoing transient exothermic chemistry. Analysis of flame propagation focuses on how the integration of thermally conductive rods ( $k = k_o \times 10^4$ ) may assist

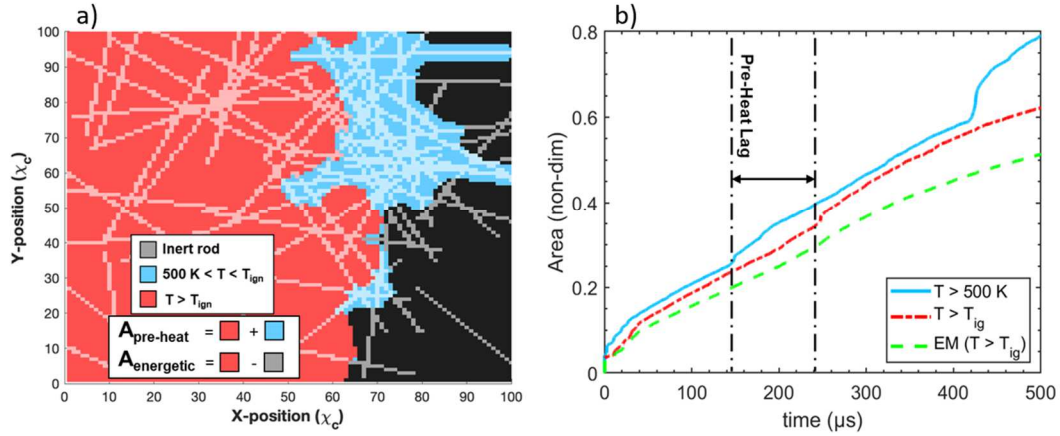


proliferation of energy throughout the composites and enhance the amount of energy released within the combustion time interval. The energy release of the system may be expressed by Equation 8-4 which can be simplified to a dimensionless energy in Equation 8-5. The dimensionless energy release is defined as the relative change in the energy release of the composite compared to the case without rods.

$$\frac{\Delta E}{\Delta t} \approx \rho c_p \Delta T \frac{\Delta V}{\Delta t} = \rho c_p \delta (T_{ad} - T_{ig}) \frac{A_E}{\tau_{total}} \quad (8-4)$$

$$\frac{\left(\frac{\Delta E}{\Delta t}\right)_1 - \left(\frac{\Delta E}{\Delta t}\right)_o}{\left(\frac{\Delta E}{\Delta t}\right)_o} = \frac{\rho c_p \delta (T_{ad} - T_{ig}) \frac{A_E}{\tau_{total}}}{\rho c_p \delta (T_{ad} - T_{ig}) \frac{A_o}{\tau_{total}}} - 1 = \frac{A_E}{A_o} - 1 \quad (8-5)$$

The energy release of the composite is directly related to the cross-sectional burned area,  $A_E$ , of energetic material assuming that the propagation behavior is relatively invariant along the thickness of the composite. The rods are incorporated to enhance pre-heating of material ahead of the main flame front. As such, analysis of flame propagation is assessed based on the degree of pre-heating, total area burned, and energetic material burned after the primary ignition transient. The pre-heated zone and total area burned are defined as the accumulation of temperature nodes which exceed 500 K and the ignition temperature (900 K), respectively. Given that the rods are non-energetic, the area of burned energetic material is simply a subset of the total area burned. An illustration is shown in Figure 8-3a outlines each area type of interest with the distribution of rods superimposed.

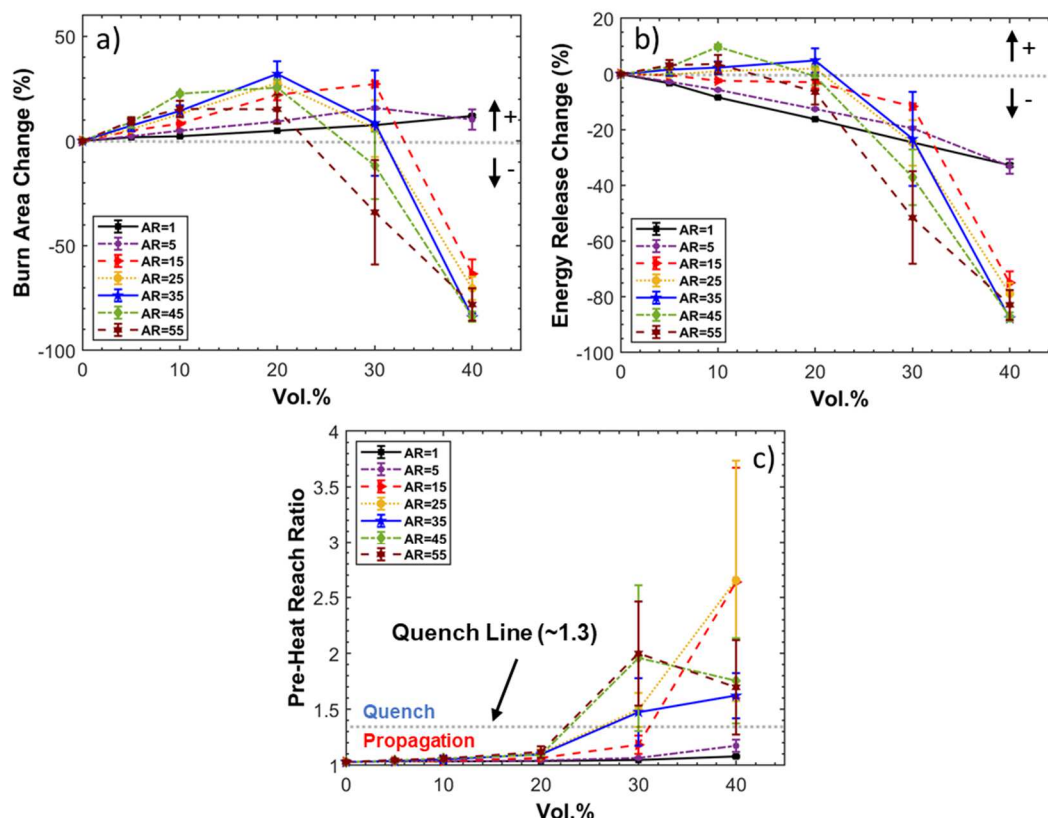


**Figure 8-3:** (a) Flame front snapshot and (b) time resolved behavior of the accumulated area of the pre-heated zone, total burned material, and burned energetic material for the case of AR=55, Vol.%=20,  $t=480 \mu s$  (left to right propagation).

As suggested, the pre-heated area is always larger than the accumulated total area burned and the burned energetic material area. One can see from the snapshot of the pre-heated zone in Figure 8-3a that the heat preferentially flows along the direction of the rods, ahead of the outlined total area burned ( $T > T_{ign}$ ). The time-resolved graph in Figure 8-3b depicts sudden jumps in the pre-heated area which corresponds to sudden increases in material conductivity due to local packing of the rods. These sudden changes are mirrored by the total area burned after a lag time corresponding to the time necessary for heat to diffuse outward from the rods to initiate chemistry.

#### 8.3.1.1 Flame propagation and energy release rates

Dimensionless comparisons of combustion performance between systems of various aspect ratios and volume percentages of rods are made in Figure 8-4 by plotting the percentage change in the total area burned,  $\frac{A_{total}}{A_o} - 1$ , and energy released relative to a composite with no rods added. Additionally, the ratio of accumulated pre-heated area relative to the total area burned is calculated to quantify how the role of pre-heating correlates to propagation behavior.



**Figure 8-4:** Effect of aspect ratio and vol.% of rods with a fixed thermal conductivity on (a) the change in total area burned, (b) energy released over the allotted burn time, and (c) the extent of pre-heating during flame propagation.

The total area burned acts as surrogate for the flame propagation rate (Figure 8-4a) where it can be seen that the total area burned is enhanced by up to 25% for most rod aspect ratios up to ~30 vol.%. Interestingly, the region of maximum enhancement between 10-30 vol.% appears to correlate to the range of volume percentages for which previous studies using carbon-based fillers (e.g. graphite, carbon nanotubes, carbon fibers) have experimentally observed the much-debated thermal percolation behavior.<sup>148,226,237</sup> However, further increases to the vol.% of added rods results in a steep regression in total area burned as a result of reduced vol.% of energetic material. Increasing the aspect ratio of rods appears to enhance the total area burned and results

in the onset of regressive behavior at lower vol.% due to further isolation of energetic material regions.

The energy release of the system in Figure 8-4b mirrors that of Figure 8-4a in terms of the general behavior with respect to vol.%. However, the energy released has a maximum enhancement of ~10% and a clear shift in the vol.% region which constitutes a positive enhancement. The 20 vol.% threshold marks the point in most of the ARs studied where replacing energetic material with thermally conductivity material is no longer beneficial for energy release. The addition of thermally conductivity materials is also not guaranteed to be beneficial with the lower half of ARs tested ( $AR < 25$ ) resulting in either negligible enhancements or reductions in energy release for vol.%=5, 10.

It appears that positive energy enhancement occurs for larger ARs and a range of low volume percentages (~5-10 vol.%). There also seems to be a threshold where increases in AR no longer increase energy release rates. In Figure 8-4a-b, the trend of increasing aspect ratio with increased energy release and total area burned regresses once the aspect ratio is increased to  $AR=55$ . When evaluating even larger aspect ratio fillers, it becomes clear that the specific orientation of the rods becomes more important at lower vol.% since the likelihood of the rods pointing in a direction conducive for flame propagation diminishes. This behavior manifests itself in both the diminished enhancement to the total area burned and energy release as well as the standard error. As a result, there must exist a unique domain of volume percentages and aspect ratios for which the energy release rate and propagation rate are maximized for different energetic systems with randomly oriented rods of a specific thermal conductivity.

Combustion experiments studying directly biased thermally conductive pathways in energetic materials are limited in scope, but have shown to significantly enhance the effective thermal conductivity in two-phase systems.<sup>176,236</sup> Modeling flame propagation using directionally manipulated rods is reserved for future works.

#### 8.3.1.2 Pre-heating of the flame front

The largest magnitude in the standard error for quantified values occurs at 30 vol.% in Figure 8-4a-b. This large error indicates a “do or die” region where there is nearly equal probability for flame propagation to either proceed or to quench. This behavior is linked to the competition between pre-heating which assists propagation, and pre-heating which leeches the heat needed to sustain self-propagation. The precise behavior of pre-heating is likely dependent on specific thermochemical properties, but the values of the ratio of the pre-heated area ( $T > 500\text{ K}$ ) to the total area burned ( $T > T_{ig}$ ) can offer insight.

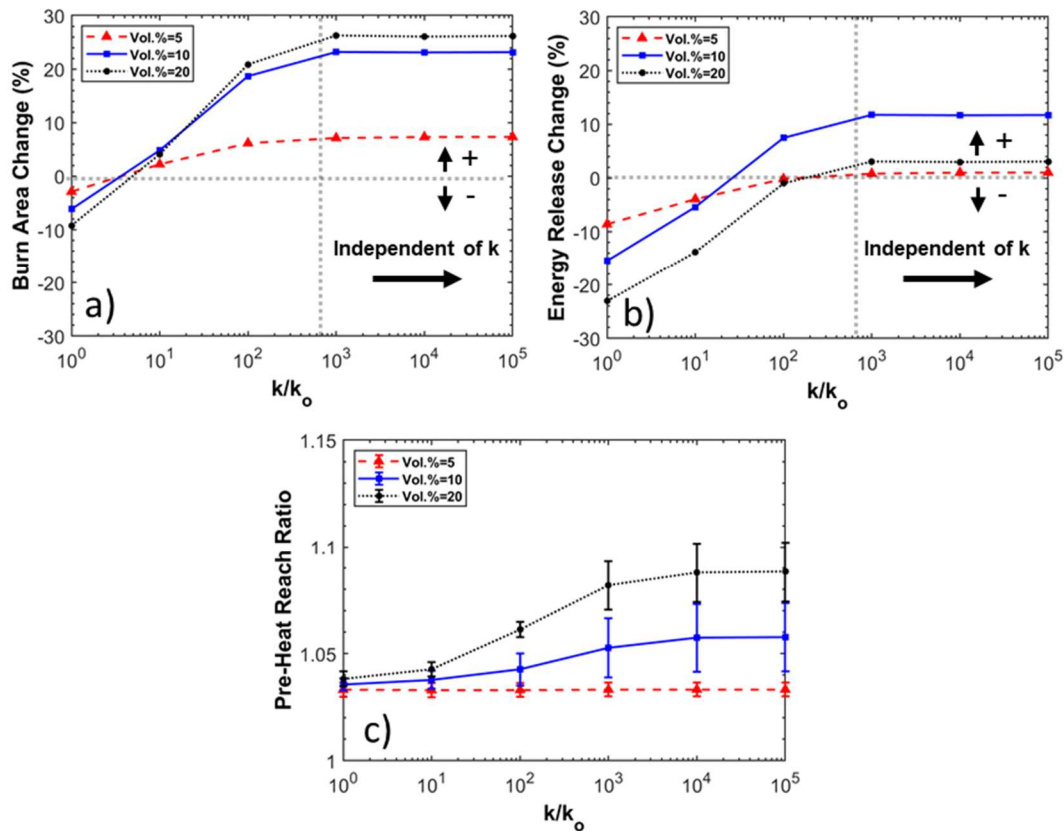
The representative pre-heat ratio for a single propagation run is determined by calculating the ratio at each time step and averaging the values over the last two-thirds of the total time steps. This is done in order to neglect the transient behavior of the initial heating process, and effects due to propagation which nears the boundary of the material (Figure 8-10). Figure 8-4c shows the reach of pre-heating increases as the vol.% increases for vol.% up to 30%. Additionally, both the values of the pre-heat ratio and the corresponding standard error at 30 vol.% clearly increases at higher aspect ratios. Increases in standard error and the pre-heat ratio express competing trends in which increases in aspect ratios may lead to propagation rate enhancements, but also may lead to increased likelihood of quenching. This trend breaks down at 40 vol.%

since it is nearly assured that higher aspect ratios will quench before establishing a self-propagating flame front.

One may roughly define a “quench line” as a threshold pre-heat value for which the probability of propagation quenching meets or exceeds the probability of sustained flame propagation. The exact position of this line is not precise but can be estimated based on the sensitivity of the standard error with respect to the collective aspect ratios as the pre-heat ratio value increases. This pre-heat ratio value of the quench line was found to be  $\sim 1.3$  and corresponds to an accumulated pre-heat area which is roughly 30% larger than the accumulated total area burned.

### 8.3.2 Role of rod thermal conductivity in flame propagation

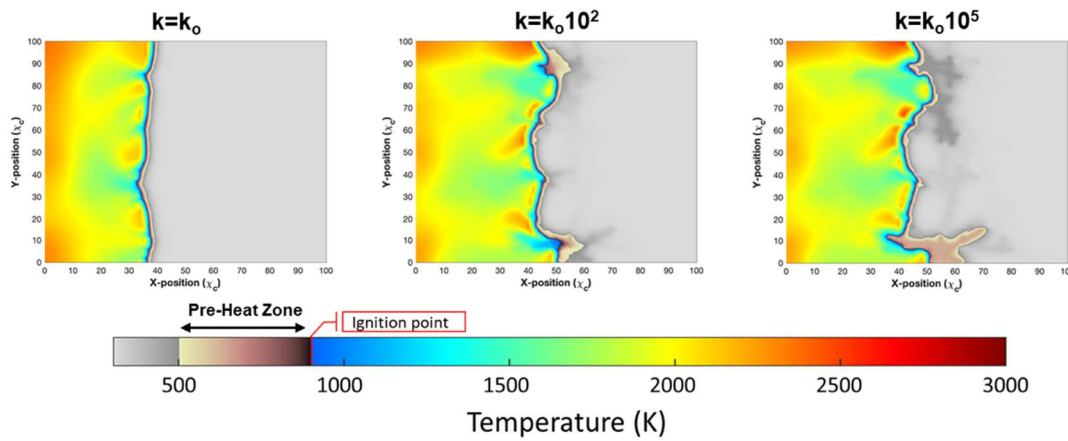
Inert rods are integrated into the energetic material to take advantage of their thermally conductive properties. The role of the thermal conductivity of rods with a constant aspect ratio ( $AR=45$ ) on the total area burned and energy released over the total combustion time is analyzed. Each volume percentage ( $vol.\%=5, 10, 20$ ) uses the same material mapping with varied rod thermal conductivity in order to focus on how the thermal conductivity, specifically, effects the total area burned and energy released. Figure 8-5 shows how the combustion behavior changes at different volume percentages as the thermal conductivity of the rods is varied by 5 orders of magnitude relative to the energetic material ( $\frac{k}{k_o} = 1, 10, 10^2, 10^3, 10^4, 10^5$ ). An example of the flame behavior with varied rod thermal conductivity at 20 vol.% is rendered in the thermal mappings displayed in Figure 8-6.



**Figure 8-5:** Effect of thermal conductivity and vol.% of rods with an aspect ratio (AR=45) on (a) the change in total area burned, (b) energy released over the allotted burn time, and (c) the extent of pre-heating during flame propagation.

It is once again observed that, while the total area burned is maximized for 20 vol.% rods, the energy release is maximized for 10 vol.% of rods. The extent of pre-heating is greatest at higher volume percentage and higher thermal conductivity. The standard error of the pre-heat ratio is calculated with respect to its fluctuations as a function of time (Figure 8-10) and is shown to increase with higher rod thermal conductivity. Although none of the parameters chosen resulted in quenching of the flame front, it can be inferred that higher thermal conductivity would result in an increased probability of quenching at higher vol. %. What is most interesting, however, is that all calculated values in Figure 8-5 reach a point where their behavior becomes

effectively independent of the thermal conductivity of the filler. With our chosen parameters, this thermal conductivity threshold occurs around  $\frac{k}{k_o} = 10^3$ . The thermal mapping of the flame front in Figure 8-6 illustrates how little total area burned is gained between  $10^2$  and  $10^5$ . With this perspective in mind, the search for sophisticated materials with exceeding large thermal conductivities may appear moot for this application.



**Figure 8-6:** Thermal mapping of energetic material propagation integrating rods with varied thermal conductivities within an invariant AR=45 rod mapping of 20 vol.% at  $t=325 \mu s$  (Propagation from left to right).

The thermal conductivity-independent region may manifest as the result of the effective thermal conductivity at boundaries between nodes with different materials. As mentioned in Section 0, the effective thermal conductivity is calculated based on the equivalent thermal resistance to heat flow across a boundary of discrete material domains (Equation 8-2).<sup>230</sup> While the effective thermal conductivity across the boundary for adjacent similar materials is equal to the bulk thermal conductivity, the effective thermal conductivity across boundaries of dissimilar materials can potentially be several orders of magnitude smaller. When the additive thermal conductivity is



much larger than the thermal conductivity of the energetic material ( $k \gg k_o$ ), it becomes clear that there is an asymptotic behavior of the amount of heat which is able to flow across the boundary as rod thermal conductivity increases. The limiting step of flame propagation becomes reliant on the thermal conductivity of the energetic material itself.

#### **8.4 Conclusions**

In this study, a 2D finite element model is employed to evaluate the role of thermally conductive, high aspect ratio additives on propagation behavior in condensed phase energetic composites. Increases in the total area burned and energy released was observed when the connectivity of thermally conductive rods throughout the composites is optimized at minimal volume percentages ( $AR > 25$ ,  $vol.\% < 20$ ). In cases where the aspect ratio was too low ( $AR < 25$ ) or additive  $vol.\%$  was too high ( $> 20 vol.\%$ ), the additive either offered no enhancement due to heat transfer limitations or quenched flame propagation through excessive heat transfer to non-energetic regions. Enhancements were reduced at aspect ratios which were too high, inferring the increased importance for rod orientation. Independently varied rod thermal conductivity for the same additive distribution led to increased enhancements in total area burned and energy release up to some critical thermal conductivity where enhancements to flame propagation performance no longer depended on the thermal conductivity of the additive. Such behavior appears to originate from limitations in heat transfer introduced by boundary resistances between material types. This study serves as proof-of-concept for a method to enhance energy release by increasing heat transfer

with additives in a manner that could be readily incorporated into the energetic composite manufacturing process.

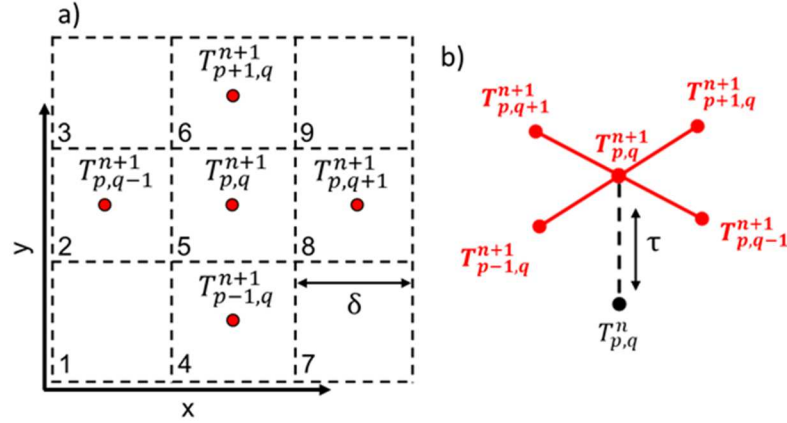
### **8.5 Acknowledgements**

This work was partially funded by the Air Force Office of Scientific Research. Reproduced from M.C. Rehwoldt\*, D.J. Kline\*, M.R. Zachariah, Numerically evaluating energetic composite flame propagation with thermally conductive, high aspect ratio fillers, Chem. Eng. Sci. 229 (2021) 116087. doi:10.1016/j.ces.2020.116087. \*Denotes that authors contributed equally to this work.

### **8.6 Supplemental Information**

#### **8.6.1 Numerical methods**

The implicit finite difference approach for this model uses the Backward Time, Centered Space (BTCS) definitions (backward Euler scheme) for the first time derivative and second spatial derivative, respectively, with corresponding truncated errors,  $O(\Delta t)$ ,  $O(\Delta x)^2$ , and  $O(\Delta y)^2$ .<sup>256,259,260</sup> Numerical solutions to the governing equation using the implicit finite difference approach are unconditionally stable in which a system of equations is simultaneously solved for all temperatures in the next time step.<sup>202,255,261</sup> Indices p, q, n represent the y-coordinate, x-coordinate, and time coordinate, respectively. Heat enters and leaves the node at each of the four boundaries outlining the grid area shown in Figure 8-7. Each temperature node is numbered based on the scheme of  $T_{p,q}^n \equiv T_{(q-1)L_x+p}^n$  where the physical grid is  $L_y \times L_x$ .



**Figure 8-7:** (a) Grid of relevant temperature nodes for position index  $p,q$  at the  $(n+1)^{\text{th}}$  time step with (b) visualization of the implicit finite difference method using adjacent temperature nodes in the previous time step.

The energy balance shown in Equation 8-6 considers only the Fourier Law conduction flux,  $\dot{q}_{cond} = -k\vec{\nabla}T$ , crossing a conduction area,  $A_c$ , over iterations of characteristic time (time step),  $\tau$ , between adjacent interior temperature nodes of density,  $\rho$ , and separated by a characteristic length,  $\delta$ . Additionally, energy is generated within the node volume,  $V$ , as a result of initial power input and chemistry. Given the volume of a node to be the cube of the characteristic length and assuming that the thermal conductivity is constant along the boundary of conduction, Equation 8-6 can be written in the more familiar Heat Equation form of Equation 8-7.

$$mc_p \frac{dT}{dt}|_{x,y} = -k_{x-\frac{\delta}{2}} \frac{\partial T}{\partial x}|_{x-\frac{\delta}{2}} + k_{x+\frac{\delta}{2}} \frac{\partial T}{\partial x}|_{x+\frac{\delta}{2}} - k_{y-\frac{\delta}{2}} \frac{\partial T}{\partial y}|_{y-\frac{\delta}{2}} + k_{y+\frac{\delta}{2}} \frac{\partial T}{\partial y}|_{y+\frac{\delta}{2}} + (P + Q)|_{x,y} V \quad (8-6)$$

$$\rho c_p \frac{dT}{dt}|_{x,y} = \frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) |_{x,y} + \frac{\partial}{\partial y} \left( k \frac{\partial T}{\partial y} \right) |_{x,y} + (P + Q)|_{x,y} \quad (8-7)$$

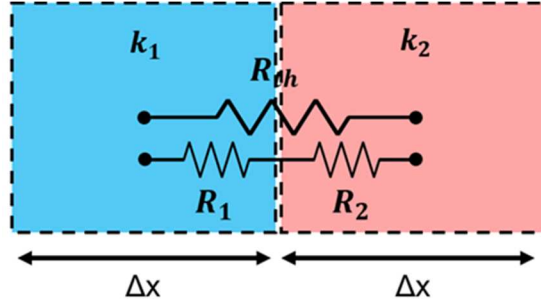
Initial input power,  $P$ , can be applied at any location until either a certain amount of time has lapsed or until the desired pre-designated temperature is reached. Chemical energy generation,  $Q$ , is the primary means of supplied energy leading to self-propagation of the flame front. A timer is started in each node once it exceeds  $T_{ig}$

so that chemistry ceases after the time allotted for chemistry ( $t_{rxn}$ ) has elapsed (Heaviside Function). For this case study, a fixed time step of 40 ns is used for which 600 chemistry time steps ( $\tau_{rxn} = 24 \mu s$ ) are allotted for each chemistry event. These values are rough orders of magnitude and are not necessarily specific to any one system. Reaction times of energetic systems typically range from 10-100  $\mu s$  or more.<sup>47,81</sup>

The thermal conductivity is spatially dependent in a discrete manner as illustrated by the 3D step-function in Figure 8-2a-b. One must consider, numerically, the change in thermal conductivity as a function of position as well as defining the thermal conductivity at node boundaries. For second order spatial accurate discretization, the change in heat conduction as a function of spatial coordinates in Equation 8-7 can be written as shown in Equation 8-8.

$$\frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) \Big|_x = k \frac{\partial^2 T}{\partial x^2} + \frac{\partial k}{\partial x} \frac{\partial T}{\partial x} \cong \frac{k_{p,q+\frac{1}{2}}^n + k_{p,q-\frac{1}{2}}^n}{2} \frac{T_{p,q+1}^{n+1} - 2T_{p,q}^{n+1} + T_{p,q-1}^{n+1}}{\Delta x^2} + \frac{k_{p,q+\frac{1}{2}}^n - k_{p,q-\frac{1}{2}}^n}{\Delta x} \frac{T_{p,q+1}^{n+1} - T_{p,q-1}^{n+1}}{2\Delta x} \quad (8-8)$$

The change in the conductive heat flux along the y-direction may be written in a similar fashion to Equation 8-7 where  $\Delta x = \Delta y = \delta$  represent the characteristic length scale of the grid point with the number of grid points equal to  $(\frac{L_x}{\delta})(\frac{L_y}{\delta})$ , where  $L_x$  and  $L_y$  represent the total width and length of the grid. Defining an effective local thermal conductivity at the boundaries (half-integer points) of discrete material regimes is accomplished by using the idea of thermal resistance.<sup>202,224,230,248</sup> The thermal resistance is analogous to electrical resistance in which the thermal resistance between two nodes is equal to the sum of the thermal resistance from one node to the boundary and the boundary to the second node,  $R_{th} = R_1 + R_2$ , as Figure 8-8 illustrates.



**Figure 8-8:** Thermal resistance approach for defining effective thermal conductivity at the boundary of discrete thermal conductivity distribution.

The effective thermal conductivity at each boundary can then be derived and defined by Equation 8-9.

$$k_{eff} = \frac{2k_1k_2}{k_1+k_2}, \quad k_{p,q+\frac{1}{2}}^n = \frac{2k_{p,q+1}^n k_{p,q}^n}{k_{p,q+1}^n + k_{p,q}^n} \quad (8-9)$$

The left side of Equation 8-7 is discretized by taking the product of the backward time definition of the time derivative, the density, and specific heat of the node in question (Equation 8-10).<sup>202</sup> The characteristic time value,  $\tau$ , is linked to the chosen characteristic length in order to maintain the level of accuracy in the model ( $\frac{\tau}{\delta^2} = const.$ ).

$$\rho c_p \frac{dT}{dt}|_{x,y} \cong (\rho c_p)_{p,q}^n \frac{T_{p,q}^{n+1} - T_{p,q}^n}{\Delta t} \quad (8-10)$$

The implicit method is implemented such that  $T_{p-1,q}^{n+1}$ ,  $T_{p+1,q}^{n+1}$ ,  $T_{p,q-1}^{n+1}$ ,  $T_{p,q+1}^{n+1}$ , and  $T_{p,q}^{n+1}$  are collected and solved simultaneously with a system of equations over the entirety of the temperature grid. The system of equations is represented in matrix form by the general equation  $A\vec{x} = \vec{b}$  where  $\vec{x}$  is the column vector representing the  $T^{n+1}$  grid temperatures,  $\vec{b}$  is the column vector representing the sum of the  $T^n$  temperatures and temperature increases,  $\beta^n$ , as a result of volumetric energy inputs, and matrix  $A$  which maps  $\vec{x}$  to  $\vec{b}$ . The ordering of the temperature nodes within  $\vec{x}$  and  $\vec{b}$  is with respect

to the numbering scheme shown in Figure 3 where the physical 2D grid is reshaped into a column vector of  $L_x L_y$  elements. As a result, matrix  $A$  must reflect this reshaping where each row of the matrix elements is representative of each equation in the system when applied to the matrix algebra. The  $L_x L_y \times L_x L_y$  square matrix is varied between time steps.

Significant changes in thermal conductivity, density, and/or heat capacity of a node as a result of phase or chemical changes (e.g. melting or final products) may be considered by establishing time/temperature dependent logic which is independent of the matrix math but reflected by the evolution of the matrix elements of  $A$  through each time step of the calculation. This may be visually represented by the dynamic behavior of the previously shown thermal diffusivity mapping over each time iteration (Figure 8-2).

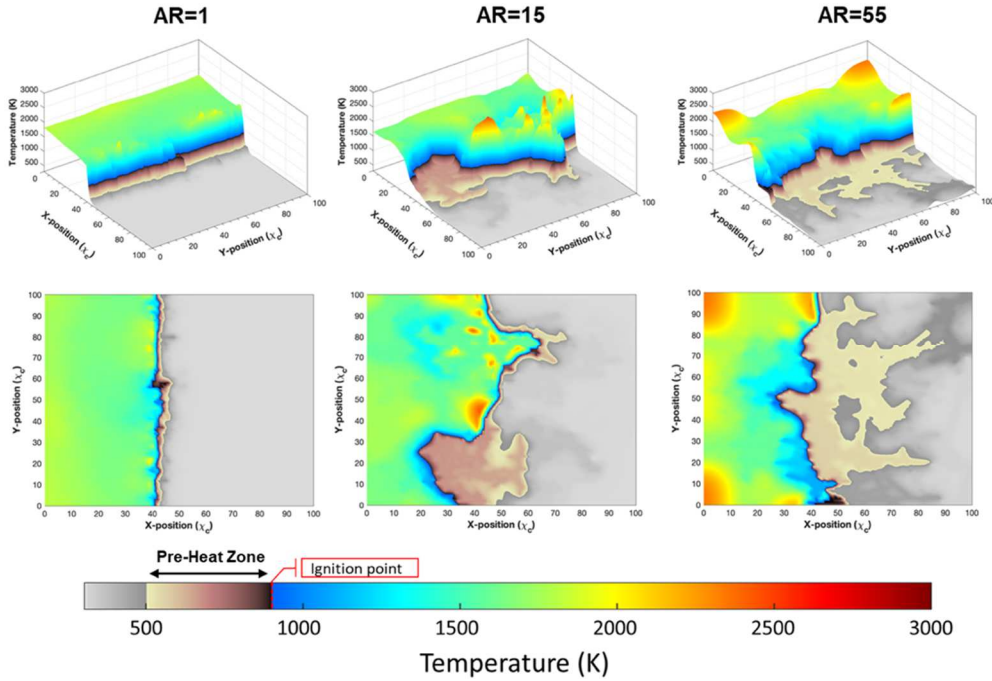
An example solution setup for a 3x3 temperature grid (Figure 8-8) with constant temperature boundary conditions is shown below in Equation 8-11. Interior node rows are those for which the central temperature node in Equation 8-11 is an interior node. The matrix elements for each interior node row represents a unique set of 5 non-zero elements  $(\lambda, \gamma, \omega, \phi, \varepsilon)$  representing the interactions between the central temperature node and 4 adjacent temperature nodes. Due to the physical grid being 3x3 in this example, there exists only one interior node represented by one central interior node row and eight boundary nodes represented by eight boundary node rows. For a general  $L_y \times L_x$  physical grid, matrix  $A$  begins with  $L_y + 1$  boundary node rows, followed by  $L_x - 2$  repetitions of  $L_y - 2$  interior node rows and 2 boundary node rows, and ending with  $L_y - 1$  boundary node rows. When considering alternative boundary conditions,

such as convective boundaries, the boundary rows will in general have unique matrix elements which emulate that of the interior node rows but with less non-zero elements depending on the boundary type (e.g. corner(3) or edge(4)). One may solve for  $\vec{x}$  using Equation 8-12 and reshape the solution back to the physical grid representation.

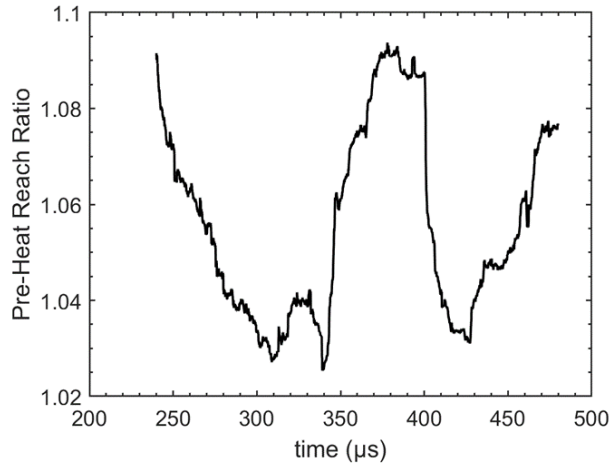
$$A_{9 \times 9} \vec{x}_9 = \vec{b}_9 \rightarrow \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & \gamma & \omega & \varphi & 0 & \varepsilon & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} T_1^{n+1} \\ T_2^{n+1} \\ T_3^{n+1} \\ T_4^{n+1} \\ T_5^{n+1} \\ T_6^{n+1} \\ T_7^{n+1} \\ T_8^{n+1} \\ T_9^{n+1} \end{pmatrix} = \begin{pmatrix} T_1^n + \beta_1^n \\ T_2^n + \beta_2^n \\ T_3^n + \beta_3^n \\ T_4^n + \beta_4^n \\ T_5^n + \beta_5^n \\ T_6^n + \beta_6^n \\ T_7^n + \beta_7^n \\ T_8^n + \beta_8^n \\ T_9^n + \beta_9^n \end{pmatrix} \quad (8-11)$$

$$A\vec{x} = \vec{b} \rightarrow \vec{x} = A^{-1}\vec{b} \quad (8-12)$$

### 8.6.2 *Additional figures*



**Figure 8-9:** Thermal mapping of energetic material propagation incorporating rods of constant thermal conductivity ( $k=k_0 10^4$ ) with varied aspect ratios at 30 vol.% at  $t=325 \mu s$  (Propagation from left to right).



**Figure 8-10:** Truncated fluctuations of the extent of pre-heating as a function of time during flame propagation.

**Table 8-1:** Table of parameters.

Physical Property (SI units)	Symbol
Thermal Diffusivity ( $\text{m}^2\text{s}^{-1}$ )	$\alpha$
Temperature (K)	$T$
Initial Input Power ( $\text{Wm}^{-3}$ )	$P$
Chemical Energy Input ( $\text{Wm}^{-3}$ )	$Q$
Characteristic time (s)	$\tau$
Characteristic length (m)	$\delta$
Aspect Ratio (Dimensionless)	$AR$
Thermal Conductivity ( $\text{Wm}^{-1}\text{K}^{-1}$ )	$k$
Input Energy Temperature Rise (K)	$\beta$
Heat Flux Area ( $\text{m}^2$ )	$A_c$
Area of Energetic material burned ( $\text{m}^2$ )	$A_E$
x-position index (Dimensionless)	$p$
y-position index (Dimensionless)	$q$
time index (Dimensionless)	$n$
Specific Heat ( $\text{JKg}^{-1}\text{K}^{-1}$ )	$c_p$
Mass (Kg)	$m$
Mass Density ( $\text{Kgm}^{-3}$ )	$\rho$
Frame Length (m)	$L$
Length Variable (Dimensionless)	$\chi_c$
Time Variable (Dimensionless)	$t_c$
Convection Heat Transfer Coefficient ( $\text{Wm}^{-2}\text{K}^{-1}$ )	$h$
Chemical Conversion value (Dimensionless)	$\eta$



## Chapter 9: Spatially focused microwave ignition of metallized energetic materials

### ***9.1 Summary***

This study investigates the ability to locally ignite metallized propellants via microwave absorption. Metallized energetic composite films incorporating high mass loadings of aluminum and titanium nanoparticle fuels within a polyvinylidene fluoride (PVDF) polymer matrix were constructed via direct write additive manufacturing (3D Printing). Simulations of power absorption for both Ti and Al nanoparticles reveal that the passivating shell composition likely plays a significant role in the observed ignition phenomenon. Various architectures of interest were constructed for predictable microwave ignition and propellant propagation. It was found that, although aluminum nanoparticles and composites do not ignite via exposure to microwaves, titanium nanoparticles can be used as an efficient reactive microwave susceptor enabling a localized ignition source. This approach enables various architectures of previously studied high energy Al/PVDF systems to be fabricated and outfitted with a microwave sensitive titanium composite in strategic locations as a means of remote ignition for aluminum systems.

### ***9.2 Introduction***

Nano-energetic materials are a class of exothermic heterogeneous solid mixtures with common applications in propellants, ignitors, and pyrotechnics. Nano-energetics seek to enhance reactivity by incorporating nanoscale metallic particles for reduced diffusion length scale between fuel and oxidizer of high combustion enthalpy redox

reactions. Although aluminum nanoparticles are the most commonly used nano-metal in energetic research studies due to its high reaction enthalpy per gram and low cost, other energetic nanoscale metals such as tantalum<sup>262</sup>, and titanium<sup>55</sup> have also been researched in past studies for their niche physical and chemical properties aside from their theoretical reaction enthalpy. Energetic formulations for propellant applications commonly integrate nano-powder within a polymer binder which itself may also act as either a fuel, as in the case of traditionally used Hydroxyl-terminated polybutadiene (HTPB)<sup>263</sup>, or an oxidizer, as in the case of fluorinated polymer binders such as polyvinylidene fluoride (PVDF)<sup>244</sup> or Viton.<sup>77</sup> Recent research utilizing metallic formulations of this type have favored a variety of additive manufacturing techniques such as layer-by-layer deposition<sup>76</sup> and stereolithography<sup>241</sup> over traditional casting methods for the prospect of fast prototyping of highly customizable architectures with tunable parameters.<sup>264</sup>

Coupled with efforts for enhanced customization of 3D printed propellants, research has been devoted to probing the ability to throttle both combustion and ignition behavior of propellants remotely by irradiating energetic materials with microwave energy (MW energy).<sup>265,266</sup> Microwave radiation is defined by the frequency range of 300 MHz-300 GHz and is utilized under low powers (~1mW) for communications and relatively higher power (~1kW) for dielectric heating of foods.

While the use of microwaves for heating has seen some success in applications for powder ceramic sintering<sup>267,268</sup>, synthesis of materials<sup>269-271</sup>, and demonstrating the initiation of crystalline high explosive monopropellant materials such as 1,3,5-Trinitro-1,3,5-triazinane (RDX)<sup>272</sup>, microwave heating of metals and metal composites has had

limited utility. The relatively high electrical conductivity of metals results in optically dense materials with skin depths of a few microns at conventional microwave oven frequencies (2.45 GHz) means that most of the electromagnetic radiation is reflected at the material surface.<sup>273,274</sup> Modeling and experimental studies of metal powders demonstrated that a remedy for this problem is to use metallic powders which couple strongly to the magnetic field, as well as having particle diameters which are of the same order of magnitude as the skin depth for that material.<sup>275–277</sup> Particles that are too large will only heat at the surface, while particles much smaller than the skin depth will absorb only a fraction of the transmitted radiation in the form of heat generated by induced eddy currents as a result of the oscillating magnetic field.<sup>275–277</sup> As such, many experimental studies focusing on heating often utilize micron size powders which may also contain magnetic materials such as iron, cobalt, or nickel.<sup>269,277–279</sup>

Although sintering of powders of magnetic metals as a result of microwave heating to high temperatures has been realized, these magnetic materials have poor energetic performance and often take several minutes (10-30 mins) to sinter.<sup>268,269</sup> Additionally, the enhancement of microwave absorption of size optimized aluminum particles ( $\sim 3\text{-}4\ \mu\text{m}$ ) showed that the temperature to which the aluminum powder reaches from microwave irradiation is not enough to ignite the sample.<sup>275</sup> A follow-on study by Crane et.al. looking at microwave heating of aluminum-iron oxide thermites demonstrated that the addition of microwave susceptors at  $\sim 10$  weight percent (wt. %), in the form of carbon-based materials, will enhance heating by microwave irradiation.<sup>280–282</sup> However, this same study concluded that, while there was an enhancement of maximum temperature reached, the maximum reach of heating which

still fell short of the ignition threshold while also hindering reactivity and flame propagation speeds. Meir et.al. have been successful at igniting micron sized aluminum-iron(II,III) oxide (magnetite) thermites within several seconds using a low power (~100W), localized solid-state microwave-drill for enhanced microwave coupling and confined heating.<sup>279</sup> This approach has shown to be successful for this specific system, however, it is unclear whether this method will work for nanoscale systems with oxidizers which may not be magnetically sensitive.

This study investigates the ability to rapidly initiate energetically relevant metallized nanoscale compositions via efficient microwave coupling, while maintaining a high overall energy density, reactivity, and combustion performance. Preliminary microwave heating tests in air of prospective fuels showed that both micropowder and nanopowder aluminum did not demonstrate thermal ignition as a result of microwave heating. However, nanopowder titanium did demonstrate thermal ignition despite the aforementioned difficulties of using nanoscale non-magnetic metals in microwave heating.<sup>276,277</sup> As a result of this initial finding, this study analyzes this behavior more in depth as well as employs the use of additive manufacturing techniques in order to design composition architectures which retrofit previously studied propellants with microwave sensitive nanoscale titanium compositions to realize a spatially localized microwave-ignitable propellants. This work also sets the stage for fabrication of tailorable composite propellants that can be precisely and remotely ignited or staged by excitation of microwave sensitive regions.

### **9.3 Methods**

#### **9.3.1 Mie theory calculations**

Calculations of the absorbed microwave intensity by an ensemble of metal nanoparticles was modeled using Mie theory which considered both bare metal spherical nanoparticles and metal nanoparticles with an oxide shell. Considering the features of the microwave source (2.45 GHz) discussed in the above section, it has been assumed that a needle antenna uniformly irradiates a circular area of 25  $\mu\text{m}$  diameter on the sample surface with electromagnetic power deliverance of 15 W. The absorption cross-section,  $\sigma_A$ , for the bare metal particles and the coated metal particles is computed using BHMIE and BHCOAT subroutines respectively.<sup>283</sup> Due to the lack of studies regarding core-shell nanoparticle microwave absorption, a more detailed study of the mechanism of microwave absorption of core-shell systems has been performed in tandem with this article by Biswas et al. which has been generalized for a wider variety of materials.<sup>284</sup> The findings of this work with respect to microwave heating of core-shell nanoparticles are applied to the materials in question and discussed in section 9.4.1.

#### **9.3.2 Materials**

Candidate materials studied in this project were determined based on three criteria: (1) high energy density nanoparticles, (2) the ability to act as an effective fuel for combustion, and (3) commercial availability with respect to cost. To meet the demands for high energy density, metallic nanoparticles were of primary interest due to their significant transient energy release during oxidation in comparison to other

potential fuels (such as carbon). In particular, aluminum and titanium were considered as metal fuels for reasons that are discussed in a later section.

Aluminum nanoparticle powder (nAl) (~80 nm diameter, ~80% active by mass) (Figure 9-7a) used in this study was purchased from Novacentrix and titanium nanoparticle powder (nTi) (30-50 nm diameter, ~70% active by mass) was purchased from U.S. Research Materials, Inc (Figure 9-7b). Nano-metals were used as received with the active content, unoxidized metal content, of the nanoparticle core-shell structure determined using thermogravimetric analysis (TGA) from previous studies.<sup>55,285</sup> Two different sizes of micron aluminum powder, 3-4.5  $\mu\text{m}$  (Figure 9-8a) and 10-14  $\mu\text{m}$  (Figure 9-8b) diameter, were purchased from Alfa Aesar and tested as-received for preliminarily microwave absorption experiments. Poly(vinylidene fluoride) powder (PVDF, Molecular Weight=534000 gm/mole) was purchased from Sigma Aldrich (Millipore Sigma) and N, N-dimethylformamide (DMF 99.8%) solvent was purchased from BDH chemicals. All chemicals were used as received.

### 9.3.3 Film characterization

Energetic composites were fabricated using a System 30M pressure-driven 3D printer purchased from Hyrel 3D. An energetic precursor is utilized as a 3D printable ink in a direct-write, layer-by-layer additive manufacturing approach to create customized multi-composite architectures with distinct ignition and propagation zones along the structure of the propellant. Each precursor 3D printable ink considered was formulated by dissolving a constant 70 mg/ml PVDF in DMF and adding various mass loadings of metal content in a similar manner to past studies of such materials.<sup>34,77</sup>

Individual mixtures and particle loadings are listed in Table 9-1 where the specific loadings of fuel and oxidizer are formulated with various equivalence ratios,  $\Phi$ , for one or more of the following reasons: stoichiometric conditions<sup>285</sup> for maximized energy output, ease of printability, enhanced propagation speed, and the optimization for microwave ignition sensitivity. Precursor suspensions are sonicated for 1 hour for each component added, then stirred for 24 hours using a magnetic stir plate/bar. The additive manufacturing technique is solvent-based and relies on the evaporation of DMF which was aided by maintaining the print bed at 70°C. The drying process is controlled by the temperature of the printing bed, the speed and extrusion rate of the print, and solvent exhaust venting. Dry times were on the order of 15-30 seconds after deposition with the resulting 4-layer film thickness being 30-40  $\mu\text{m}$  (7-10  $\mu\text{m}/\text{layer}$ ) as shown in Figure 9-9. A minimum of four layers of printed material is required to effectively remove the samples from the printing surface. Films for flame speed and ignition tests are printed in rectangular sheets from which multiple free-standing thin strands (1 mm x 2 cm) may be harvested. More complex architectures are printed using multiple printing codes designated for specific formulations which work in tandem to complete the final multi-component architecture. Multi-component architecture prints have layers of different formulations printed directly on top of a previous layer in a partial-layering scheme. Figure 9-9 depicts cross-section images of Al/PVDF and Ti/PVDF samples printed with 4 layers at  $\sim 8 \mu\text{m}/\text{layer}$ .

**Table 9-1:** Energetic precursor formulations.

Fuel Type (wt.%)	PVDF (wt.%)	Equivalence Ratio ( $\Phi$ )
Al: 25	75	1
Al: 65	35	5.3
Ti: 35	65	1

Ti: 50	50	1.9
Ti: 65	35	3.5

---

#### 9.3.4 *Burn rate characterization*

Combustion performance of printed energetics was evaluated using 1 mm-wide energetic strands which were cut from larger printed sheets and burned in a controlled atmosphere. As shown in Figure 9-10, the strands are loaded into a glass cylinder closed off with rubber stoppers that have gas inlets and electrodes. The strands are mounted between two braided nichrome wires and anchored at the opposite end with double-sided tape. Strands are ignited by resistively heating the nichrome wire and the combustion event recorded using a high-speed camera (Phantom Miro M110) recording at 10,000 frames/s. Burn rate is calculated by tracking the pixel coordinates of the flame front as a function of time and considering the pixel/distance ratio determined from the precisely spaced metal plates in the foreground of the video.

Burn rate experiments for all samples were performed in both air and argon environments to reduce the dependence of oxygen on the observed ignition and propagation characteristics. Anerobic combustion tests used argon (Airgas 99%) to purge the glass tube for ~5 minutes followed by the simultaneous closure of the inlet/outlet prior to the ignition event.

#### 9.3.5 *Focused microwave ignition*

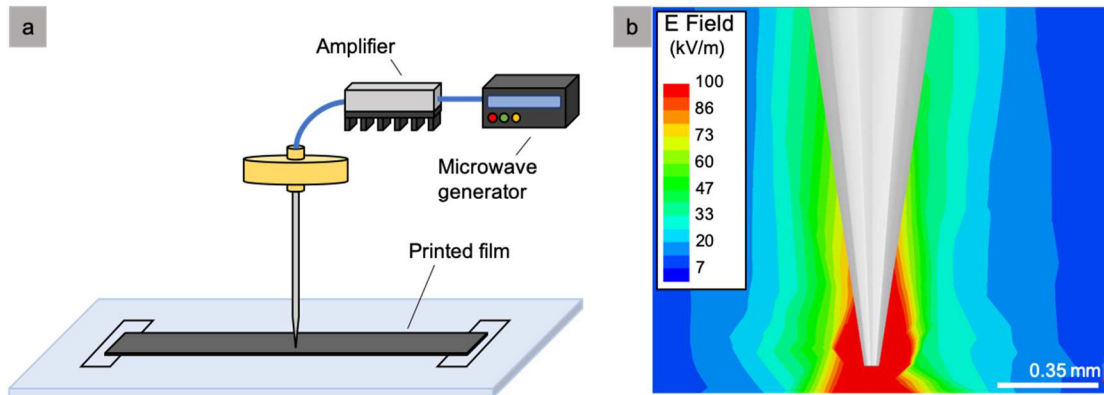
To precisely localize ignition of the microwave-sensitive energetic composites, a needle-based, coaxial microwave emitter was designed and suspended over the printed films.<sup>279</sup> A critical characteristic of this design is the ability to concentrate microwave energy into an area much smaller than the chosen wavelength. In this



configuration, a solid-state power amplifier (AMCOM AM206545SF-3H) with a peak gain of 32 dB drives a 2.45 GHz, 10 dBm continuous wave tone from a microwave signal generator (DS Instruments SG12000). This brings the peak power output of the device to approximately 42 dBm, or 15 W (see Figure 9-1). A coaxial-solder pin adapter was soldered to a nickel-plated straight pin (Singer 0.025" diameter). The printed film sample sits on a glass slide and the slide is positioned over a grounded surface. A simulation of the electric fields through Ansys HFSS illustrates the field intensity ( $\sim 100$  kV/m) is most concentrated between the pin and ground plane with the sample in its path. This field is an order of magnitude lower than the dielectric breakdown point of air at 1 atm ( $\sim 3000$  kV/m) and does not result in electrostatic discharge events.

Since the working distance from emitter to sample is much smaller than the operating wavelength, sample exposure is in the “near-field” of the electromagnetic (EM) field, implying there will be some coupling between sample and antenna. Concentrating microwave radiation in the far-field down to a point would require a large antenna design and sample size but would result in the same phenomenon and observations presented here.

Combustion events of the needle-based microwave ignition experiments were observed using a high-speed camera (Phantom Miro M110) which was triggered simultaneously with the power input to the microwave amplifier.



**Figure 9-1:** (a) Needle-based microwave ignition experiment at 2.45 GHz. Peak power of the entire system is ~15W. (b) Simulations of the electric field at the tip of the needle for a frequency of 2.45 GHz (using Ansys HFSS and an auto-generated mesh).

Samples tested in argon with the microwave needle assembly were placed inside a glass tube with gas inlets. A small sample of material was placed inside the tube, the tube was purged with argon, and the needle was then placed through an opening in the tube to access the interior for sample heating. Videos recorded used the same high-speed camera as was used in the other experiments described.

## 9.4 Results and Discussion

### 9.4.1 Microwave absorption of metals and materials selection

Efficiency of microwave heating is closely related to material properties such as electrical conductivity ( $\sigma$ ), permittivity ( $\epsilon$ ), and permeability ( $\mu$ ) when interacting with electromagnetic radiation of a certain frequency ( $\omega$ ). As such, the total averaged power dissipated ( $P_d$ ) within a material is linked to these properties in their ability to couple to the electric field ( $E$ ) and magnetic field ( $H$ ) of the incident electromagnetic radiation. A steady-state electromagnetic energy balance between the energy flux delivered by the microwave to the surface of the material, described by the Poynting

vector, and the time averaged energy dissipated per unit volume yields Equation 9-1.<sup>274,283</sup> The permittivity,  $\varepsilon = \varepsilon_o(\varepsilon'_r - j\varepsilon''_r)$ , and permeability,  $\mu = \mu_o(\mu'_r - j\mu''_r)$ , are generally complex with the real parts representing a measure of energy storage from the EM wave within the material and the imaginary part representing the energy dissipated within the material.<sup>273</sup>

$$P_d = \frac{1}{2}\sigma|E|^2 + \frac{1}{2}\omega\varepsilon_o\varepsilon''_r|E|^2 + \frac{1}{2}\omega\mu_o\mu''_r|H|^2 \quad (9 - 1)$$

Each term on the right side of equation 9-1 represents, from left to right, the conduction, dielectric, and magnetic heat losses respectively where the solution to the electric and magnetic field throughout the volume of the material may be solved by using the Maxwell Equations for materials.<sup>274,283</sup> The general form of this solution is a traveling plane wave with an exponential attenuation factor to represent the loss of energy of the incident radiation. The characteristic attenuation factor is related to what is colloquially known as the skin depth of a material and represents the length scale to which incident electromagnetic radiation may penetrate a material to efficiently heat.<sup>274,276,283</sup> The skin depth is dependent on the frequency of the incident electromagnetic radiation and the electrical conductivity of the material. Metals generally have a skin depth on the order of a few microns where the optimal size of particles for absorption of microwave energy is of the same order of magnitude as the skin depth.<sup>276,277</sup> Based on this information, and the previous findings mentioned in the introduction, working with nanometals should be intrinsically difficult given that nanoparticles have diameters that are 100 times smaller than the skin depth.

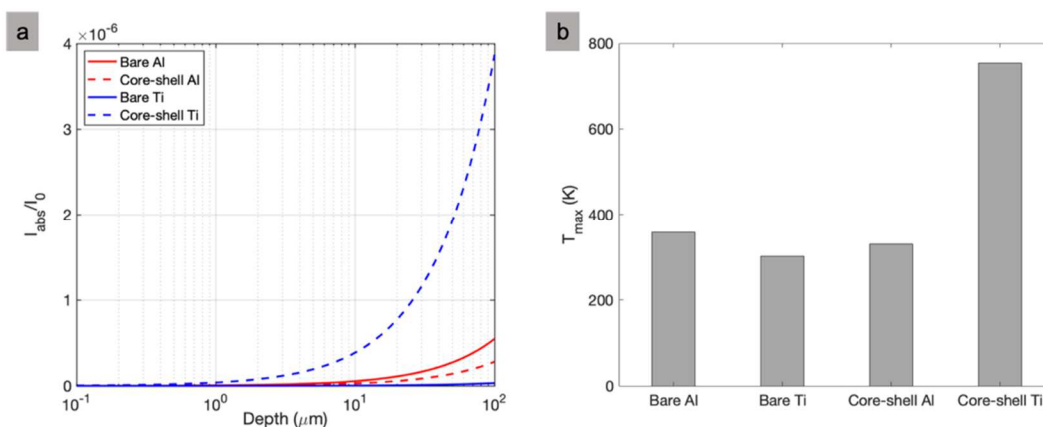
Having set the candidate criteria (as discussed in the materials section above), research began by studying the most-widely used metal fuel – aluminum. Aluminum-based energetics are well-studied in literature and incorporation into a polymer (particularly poly(vinylidene fluoride), PVDF) has proven successful in making solid propellants that can be readily manufactured.<sup>77,244</sup> Unfortunately, the initial investigations into microwave heating of Al proved unsuccessful, which led to consideration of another reactive metal as an initiator for aluminum-based composites. As this paper will demonstrate, titanium nanoparticles and titanium-based composites proved to be highly successful at igniting via microwave exposure. The reasons for its effectiveness are discussed below.

A comparison of microwave power absorption by ensembles of aluminum and titanium nanoparticles in an inert medium was conducted using Mie theory.<sup>283,284</sup> The intensity fraction absorbed is given by Equation 9-2 where  $I_{\text{abs}}$  is the absorbed intensity,  $I_0$  is the incident intensity,  $\sigma_A$  is the absorption cross-section,  $N$  is the number density of particles, and  $d$  is the path length (as characterized by the layer)<sup>286</sup>. Equation 9-2 is derived from the Beer-Lambert law, which is a valid assumption since the scattered radiation is small given that the particle size is much smaller than the wavelength of light and the scattered radiation from other particles would not interact. The effective relative complex permittivity for Ti and Al were estimated using the Drude model approximation for metals ( $\tilde{\epsilon}_r \approx \tilde{\epsilon}_r' + j \frac{\sigma}{\omega \epsilon_0}$ ).<sup>274,283,287</sup> For an operating frequency of 2.45 GHz, the approximate effective relative permittivity of highly conductive particles is dominated by the imaginary loss term,  $\tilde{\epsilon}_r''$ , which in this model is directly related to electrical conductivity and Joule heating term in Equation 9-1.<sup>277,287</sup> Lastly, since both

metals are non-magnetic, the magnetic permeability for all materials is assumed to be 1. Calculated values for all aforementioned constants can be seen in Table 9-3.<sup>273,277,288,289</sup>

$$\frac{I_{abs}}{I_0} = 1 - e^{-\sigma_A N d} \quad (9 - 2)$$

Figure 9-2a shows the resulting Mie theory fractional microwave extinction as a function of layer thickness of a composite comprised of 80 nm metal particles. Most interesting is that, even though as-received core-shell nTi is an effective absorber relative to core-shell Al in experiment (discussed in Section 9.4.2), the Mie calculation indicates otherwise for bare particles. When one considers nanoparticles, a non-trivial fraction of the mass is concentrated in the passivating shell. The shell composition of core-shell nanoparticles maybe characterized by a combination of X-ray photoelectron spectroscopy (XPS) and X-ray diffraction (XRD). Aluminum naturally forms a 2-10 nm passivating oxide (Al<sub>2</sub>O<sub>3</sub>) shell which can dramatically impact combustion performance and material properties at small particle sizes.<sup>59,290</sup> Titanium nanoparticles readily acquires a shell of similar thickness, but instead demonstrates, from XPS, a mixed molar compositions of TiN (~20% by number) and TiO<sub>2</sub> (~80% by number) when exposed to air.<sup>55</sup> Considering this difference in shell composition, previous studies regarding the microwave sensitivity of metals by other groups may have failed to consider the impact that the passivating shell may have on the interaction of the electromagnetic field with the internal pure metal contents.<sup>276</sup>



**Figure 9-2:** (a) Calculated fraction of microwave (2.45 GHz) intensity absorbed for bare and core-shell Al and Ti nanoparticles in a 25  $\mu\text{m}$  diameter cylinder as a function of depth and (b) the estimated maximum temperature after being exposed to a microwave input of 15 W for 0.5 s. Physical properties used in calculation are provided in Table S1.

Figure 9-2a shows the calculated absorbed microwave (2.45 GHz) intensity for the passivated Al and Ti nanoparticles where the inclusion of the shell significantly impacts the estimated absorbed intensity. Although the bare Al nanoparticles are expected to absorb more power than similarly sized Ti particles, the inclusion of a Ti passivating shell increases the intensity absorbed by an order of magnitude over the Al equivalent. This suggests that the shell must play a significant role in the heating of metal particles. The transparency of  $\text{Al}_2\text{O}_3$  does not enhance the absorbed fraction of the metal nanoparticles<sup>273</sup>, while the material properties of TiN and  $\text{TiO}_2$  increase the power fraction absorbed of the otherwise inferior microwave absorbing bare Ti particles.<sup>284</sup>

The material properties alone do not explain why aluminum nanoparticles are so drastically inefficient at coupling with microwaves compared to Ti nanoparticles since both particle types are exposed to the same method of heating. Models considering bare metal particles suggest that heating is dominated by induced eddy

currents from the magnetic field.<sup>276,277</sup> However, even micron sized aluminum particles cannot be efficiently heated and as particles are reduced to the nano-scale, the magnetic contribution drastically diminishes, further ruling out any significant heating from microwave absorption. Concurrent work by the Zachariah group expanding the system to a core-shell system has shown, from the Mie Theory approach, that not only is the behavior of the previous models recaptured, but it is also shown that the core-shell structure at these length scales, in combination with the material properties of the core-shell, can significantly enhance the dissipation of power through the electric field of the microwave to several orders of magnitude higher than the magnetic field contribution when at its peak.<sup>284</sup>

The maximum achievable temperature of the coated and bare metal particles was estimated using the calculated power fraction absorbed, the thermophysical properties of the contents (heat capacities retrieved from the NIST webbook and averaged over the core-shell system on a mass basis), and assuming a power input of 15 W for 0.5 s over a circular area with a  $\sim 25\text{ }\mu\text{m}$  diameter. The particles are assumed to be perfectly insulated with volumetric heating throughout for which these results would represent the upper bound of the temperature reach over the 0.5 s heating interval. When accounting for the passivating shell on the Ti and Al particles, the added power absorption contribution by the Ti shell increases the estimated temperature to  $\sim 750\text{K}$ , above the documented ignition temperature of Ti in air ( $\sim 670\text{K}$ ).<sup>55</sup> The increased power transmission provides more efficient heating of the higher energy density Ti core, leading to thermal ignition.

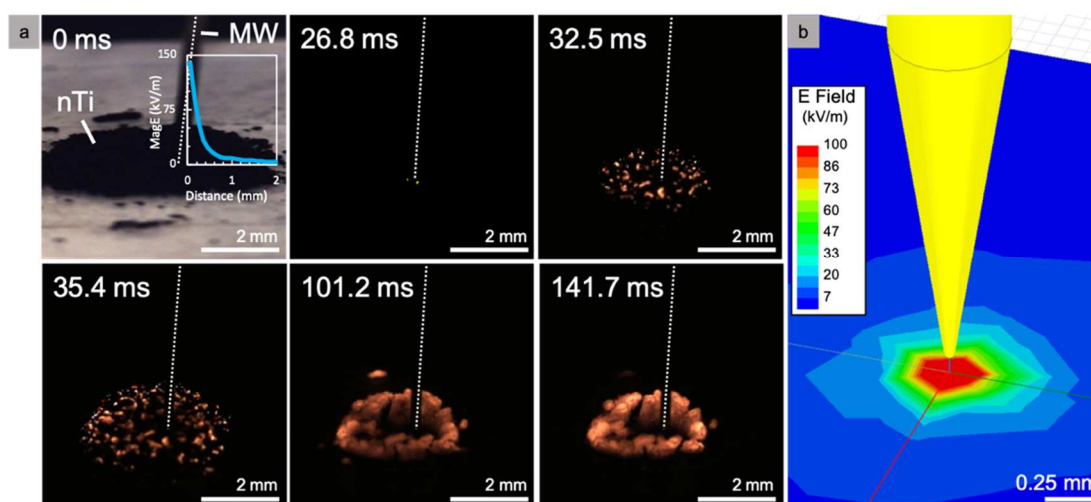
#### 9.4.2 *Microwave sensitivity of metal powders*

Potential nano-powder fuels for solid propellant precursors materials were tested in air for sensitivity to microwaves by first observing whether the bare powder would absorb enough microwave energy to be heated to sufficiently high temperatures for ignition. Consistent with previous literature, aluminum particles were unable to be sufficiently heated with microwaves, independent of the particle size.<sup>275</sup> Aluminum particle sizes ranging from 80 nm – 14  $\mu\text{m}$  were unable to ignite with microwave energy, reaffirming the poor coupling to the sample which may be hindered by the oxide shell of  $\text{Al}_2\text{O}_3$ .<sup>291</sup> In contrast to the results for the aluminum particles, titanium nanoparticles subject to the microwave needle assembly demonstrated heating phenomena within several milliseconds of the microwave amplifier being powered (see Figure 9-3). The heating event appears to originate from a single location that is approximately near the location of the maximum electromagnetic field. The heat is quickly transferred to surrounding particles from this hot spot as the Ti begins reacting with the  $\text{O}_2$  in air until the sample is uniformly heated. One important consideration in these experiments would be the impact of electrostatic discharge (ESD) (arcing) within the spacing between the particles that could transfer enough energy to start the reaction.<sup>270</sup> However, no visible anisotropic discharge events were identifiable for any of the samples tested during the ignition process.

Microwave heating of Ti nanoparticles was also performed in an inert atmosphere in order to demonstrate heating without reactive heat generation. This was done using the microwave needle assembly which was placed into an argon environment using a glass tube with gas inlets and a small opening at the top for



microwave needle access. Upon powering the microwave amplifier, the metal powders immediately surrounding the antenna began to heat to glowing temperatures within milliseconds. However, unlike the experiments performed in air, the particles would continually glow without the heat being transferred to nearby particles due to the lack of  $O_2$  and  $N_2$ , preventing the exothermic formation of  $TiO_2$  or  $TiN$  (see Figure 9-11). These results demonstrate that not only can Ti nanoparticles be heated to high temperatures using microwaves, but the high temperatures achieved are not solely dependent on the Ti reaction with air. The heating of Ti nanoparticles in air merely enhances heat generation for thermal propagation away from the localized microwave source through its strongly exothermic reaction with air. The sensitivity of the Ti particles to microwaves further alludes to the importance of shell composition on particle heating. Experimentally, core-shell Ti nanoparticles can couple to microwaves more efficiently, thus heating faster and ultimately achieve thermal ignition as opposed to core-shell Al nanoparticles which are unable to be heated via this method.



**Figure 9-3:** (a) Ignition of nano-Ti powder in air (50nm diameter) using microwave needle apparatus imaged using a high-speed camera.  $t=0$  defined as starting time for power supplied to the applicator. Dashed white line indicates position of needle. (b)

Simulation of the experiments in (a) showing the near field electric field on the ground plane performed using Ansys HFSS and an auto-generated mesh.

As mentioned previously, the observed heating and ignition of the samples by microwaves is likely attributed to (1) the unique shell composition of the Ti nanoparticles and (2) the lower ignition temperature of Ti particles in air. Electrical conductivities of materials found in the nano-Ti shell (TiN, TiO<sub>2</sub>), while having a wide range of reported values depending on crystal structure and physical architecture, are consistently larger than those for the Al<sub>2</sub>O<sub>3</sub> found on the exterior of the Al nanoparticles.<sup>273,277,288,289</sup> Microwaves couple well with Ti nanoparticles as a result of unique material properties and its core-shell structure. The entire particle is heated via Joule heating originating from the dynamic nature of electromagnetic waves, resulting in frictional heat caused by induced eddy currents and shell enhanced induced polarization from the magnetic and electric components of the microwave, respectively.<sup>276,277,284</sup>

#### 9.4.3 Microwave sensitivity of metallized propellants

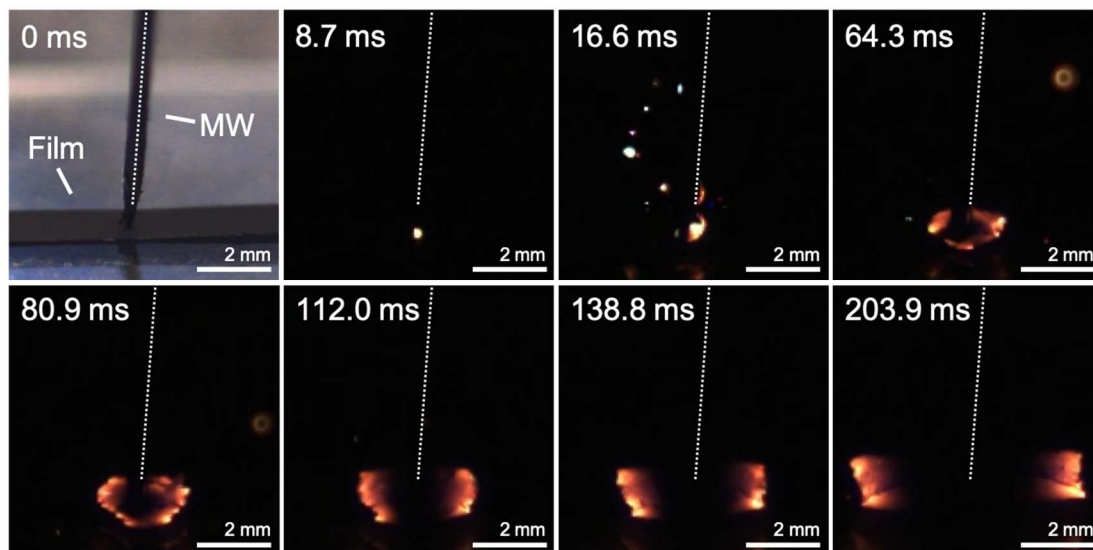
While the potential to ignite Ti-based energetic materials remotely using microwaves has been demonstrated, the applications of the mixtures tested thus far are limited in scope. Pure metal powders react rather slowly despite their high energy density. One particularly interesting application of remote ignition would be solid propellant mixtures since they have received renewed interest with the rise of additive manufacturing.<sup>76,241,264,292,293</sup> By incorporating metal particles into an energetic polymer solution, solid propellants can be easily manufactured by additive

manufacturing technologies with tailorable architectures to produce custom performances based on their physiochemical properties.

Solid energetic precursors of Ti/PVDF were prepared with varying weight percentages of Ti, as shown in Table 9-1, to investigate the role of Ti metal particles on ignition. It was expected that the addition of PVDF to the Ti particles would impact the ignition of the materials, especially considering that particles would be predominately encapsulated in the polymer throughout the film. It was observed that at an equivalence ratio ( $\Phi$ ) of 1, Ti (35 wt. %)/PVDF (65 wt. %) will not be ignited by microwave radiation, but increasingly Ti fuel rich compositions will ignite more readily and consistently. Ti (50 wt. %)/PVDF (50 wt. %) was found to be a lower bound for microwave ignitability for which heating was inconsistent at times. Additionally, the level of Ti particle inclusion must also be balanced with printability where Ti (65 wt. %)/PVDF (35 wt. %) was found to be the upper bound for Ti loading. A summary of these results is presented in Table 9-2. An important note is that electrical percolation may be contributing to the microwave sensitivity of these composite propellants, however all results and calculations presented thus far suggest that the as-received Ti nanoparticles would ignite irrespective of percolation.<sup>294</sup> A detailed analysis on percolation thresholds is beyond the scope of this article but should be considered in future research.

When tested with the needle-based microwave applicator, Ti/PVDF samples containing 65 weight-% Ti were readily ignited within 10 ms of power being delivered to the samples with the propellant films consistently propagating to completion in air (see Figure 9-4) and argon (discussed in the following section). Upon ignition of the

sample, a bright ignition point becomes clear with particles being ejected from the surface shortly after. Within 65 ms, a uniquely shaped circular reaction front with an approximate width of  $\sim 0.3$  mm began propagating from the hot spot until it reached the edge of the film at which point the propagation became linear. The flame structure of the Ti/PVDF films during combustion strongly resembles those that are seen in Al/PVDF combustion events with a long, hot reaction tail which generates a significant amount of soot, however a detailed study is reserved for future research. While the structure of the flame will be explored in future work, the bi-directional and steady propagation of the Ti/PVDF film opens up unique avenues by which to employ this material as a remote ignition source with high targetability.



**Figure 9-4:** Ignition of Ti/PVDF (65 wt.% Ti) film in air using microwave needle apparatus imaged using a high-speed camera.  $t=0$  defined as starting time for power supplied to the applicator. Dashed white line indicates position of needle.

#### 9.4.4 Controlled-environment burn characteristics of metallized propellants

Flame speed experiments of Al/PVDF and Ti/PVDF were performed in air and argon environments to evaluate the potential for implementation into a solid propellant.

Al/PVDF has been previously investigated as an additively manufactured solid propellant, however, Ti/PVDF has yet to be fully studied.<sup>34,77,285</sup> While air would likely be the environment for combustion of these materials, argon was also considered to study flame self-propagation performance within an environment devoid of both oxygen and nitrogen.

Results of the average burn velocity of four samples of Al/PVDF and Ti/PVDF at different mass loadings of metal fuel are presented in Table 9-2 where it can be seen that the burn velocity of Al/PVDF samples consistently outperforms the Ti/PVDF equivalents (raw data in Figure 9-12, Figure 9-13, Figure 9-14, and Figure 9-15). While the inert argon environments do substantially impact the burn velocities of both sample types, Al/PVDF is still able to burn at faster rates with lower metal loadings than Ti/PVDF did in air. Furthermore, low-Ti-content energetic samples were unable to sustain combustion in the inert environment. It is believed that the kinetics of the Ti/PVDF reaction in an Ar environment are not favorable since XRD analysis of post combustion products predominately show  $\text{TiF}_3$  as the primary product (see Figure 9-16) whereas one would expect the formation of  $\text{TiF}_4$ , indicating that the reaction is not going to completion. In contrast, when the Ti/PVDF samples are burned in air, the Ti also exothermically reacts with  $\text{O}_2$  which would assist in any shortcomings in the energy release due to incomplete reactions with PVDF.

**Table 9-2:** Free-standing burn velocities in air and argon environments and ability to be ignited with microwaves.

Sample	Metal wt. %	Environment	Burn Velocity	MW Ignition
Al / PVDF	25	Air	$\sim 14 \pm 1 \text{ cm/s}$	N
		Argon	$\sim 7 \pm 1 \text{ cm/s}$	N
	65	Air	$\sim 27 \pm 2 \text{ cm/s}$	N
		Argon	$\sim 15 \pm 4 \text{ cm/s}$	N

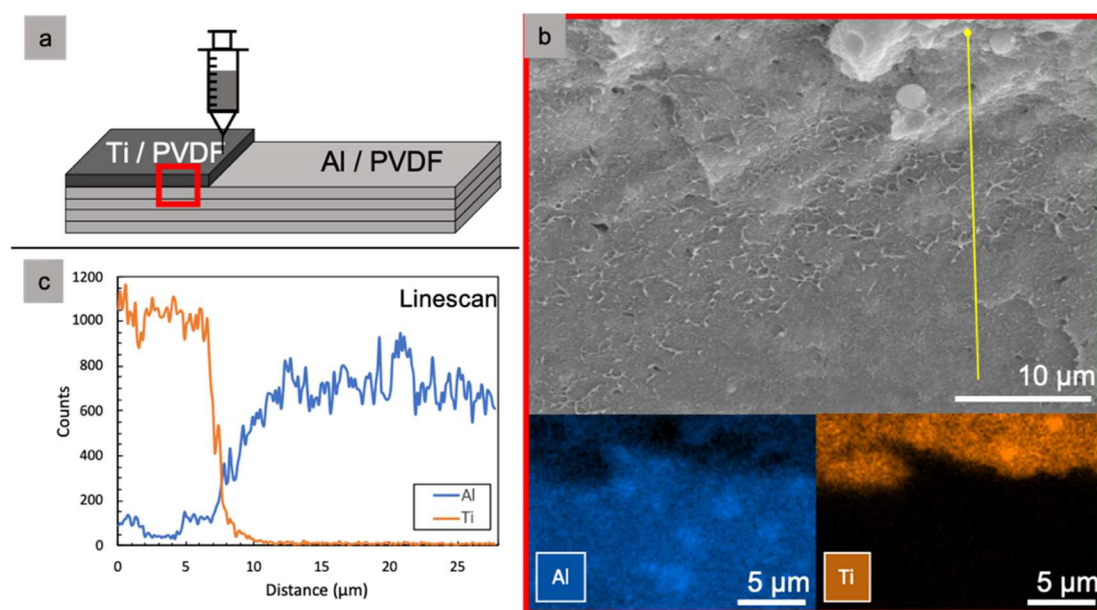
Ti / PVDF	35	Air	$\sim 1 \pm 0.5$ cm/s	N
		Argon	-----	N
	65	Air	$\sim 4 \pm 0.1$ cm/s	Y
		Argon	$\sim 1 \pm 0.1$ cm/s	Y

From these results, one can see that while both Al/PVDF and Ti/PVDF samples can propagate in both air and argon environments, the Al/PVDF samples would be more desirable as a conventional solid propellant since it would be able to generate more thrust by releasing similarly amounts of stored energy over shorter periods of time. However, the insensitivity of aluminum to ignition by microwave radiation limits the ability of the propellant to be utilized in remote-initiation situations. Considering the limitations between the propellant formulations, it would be desirable to have a microwave-sensitive propellant which does not have reduced performance. A potential architecture is explored in the following section.

#### 9.4.5 *Architecturing of a microwave-initiated solid propellant*

The results presented thus far in this work suggest that, while Ti/PVDF may not be an optimal solid propellant choice with a slow propagation velocity, it can be readily ignited in both aerobic and inert environments using targeted microwave radiation and will steadily propagate in these environments at high mass loadings of Ti. Although Al/PVDF does not have the same microwave sensitivity, its relatively superior combustion performance as a propellant and relatively low manufacturing cost still would likely steer most rocketeers toward using Al/PVDF (or some other aluminum-based metallized composite) as a primary propellant mixture.<sup>35,77,87,95,295</sup> As such, this work also investigated the potential incorporation of Ti/PVDF in an igniter application for Al/PVDF propellants that are additively manufactured.

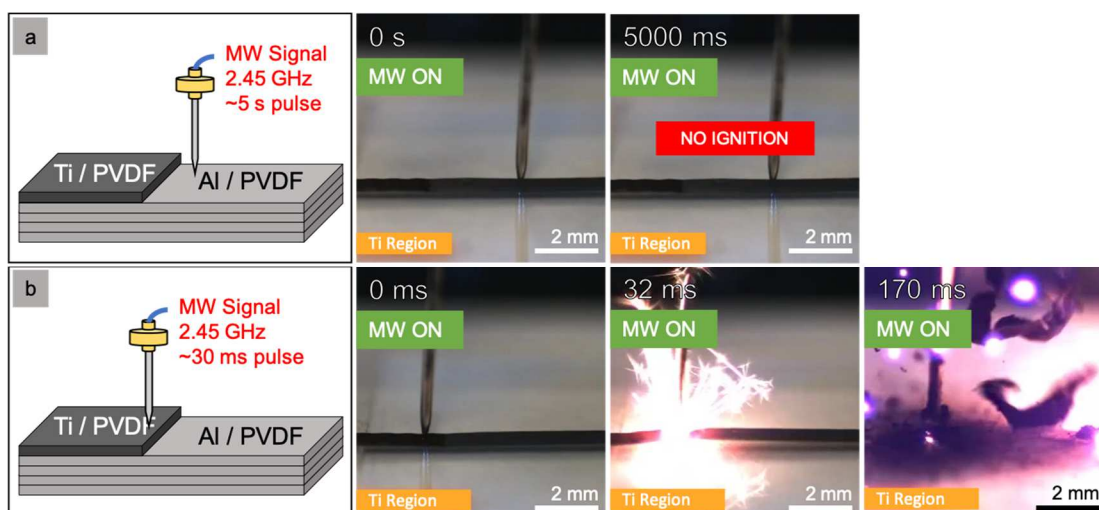
An Al/PVDF precursor ( $\Phi=1$ ) was prepared and printed 4-layers high ( $\sim 30\text{ }\mu\text{m}$ ) using a direct-write additive manufacturing method. A Ti/PVDF layer ( $\sim 7\text{ }\mu\text{m}$ ) (65 wt.% Ti) was then printed in a smaller segment on top of the Al/PVDF sample. A depiction of the sample can be seen in Figure 9-5 and scanning electron microscopy/energy-dispersive X-ray spectroscopy (SEM/EDS) images demonstrating a clear separation between the Al/PVDF and Ti/PVDF layers that were printed. A line scan of the sample originating from the Ti/PVDF sample show that the single Ti/PVDF layer is  $\sim 7\text{ }\mu\text{m}$  thick and that the Al/PVDF section is  $\sim 4\times$  thicker than the Ti/PVDF domain with a total thickness of roughly  $35\text{ }\mu\text{m}$ .



**Figure 9-5:** (a) Depiction of the architecture for a layered Al/PVDF, Ti/PVDF propellant that can be initiated with microwaves where layers 1-4 are Al/PVDF ( $\Phi=1$ ) and layer 5 is Ti/PVDF (65 wt.% Ti) ( $\sim 35\text{ }\mu\text{m}$  thick). (b) SEM/EDS image of layered film cross-section as outlined in a. (c) EDS line scan results for line drawn in b.

Combustion experiments were performed using the needle-based apparatus in aerobic and inert environments which yielded positive results for Ti/PVDF to act as a microwave-sensitive ignition source for Al/PVDF. Figure 9-6 shows that the

microwave sensitized Ti/PVDF layer printed on top of the larger Al/PVDF strand can be ignited ~30 ms after power is supplied to the microwave needle. Within 20 ms after the Ti/PVDF ignition, the exothermic reaction then transfers enough heat to the underlying Al/PVDF layers to rapidly (<1ms) reach its ignition temperature of ~620 °C, leading to sustained propagation.<sup>34</sup> Figure 9-5 and a supplementary video also demonstrates how the ignition location can be selectively placed over a microwave-insensitive Al/PVDF layer and captures the reaction propagation transition from the Ti/PVDF into the Al/PVDF.



**Figure 9-6:** Microwave needle ignition of a layered Al/PVDF, Ti/PVDF propellant where layers 1-4 are Al/PVDF ( $\Phi=1$ ) and layer 5 is Ti/PVDF (65 wt.% Ti). (a) Al/PVDF region does not ignite when exposed to microwave radiation and (b) Ti/PVDF layered region ignites and propagates into Al/PVDF region.

By making a specific segment of solid propellant remotely initiated using selectively placed microwave sensitive materials, this work sets up the development of remotely-staged, additively manufactured solid propellants. Currently, solid rocket motors do not have an “off” switch like their liquid rocket engine counterparts. Incorporating a small, inert “quenching” region with low thermal conductivity into a solid rocket motor assembly, followed by another microwave-sensitive ignition source,



could provide “checkpoints” between different stages of the rocket and even a moderate throttle control (multiple segments simultaneously) using only a directional microwave radiation source and an antenna. The design of the solid propellant could be readily implemented using more sophisticated additive manufacturing techniques and would significantly improve the versatility of solid rocket motors for civilian and military applications.

### ***9.5 Conclusions***

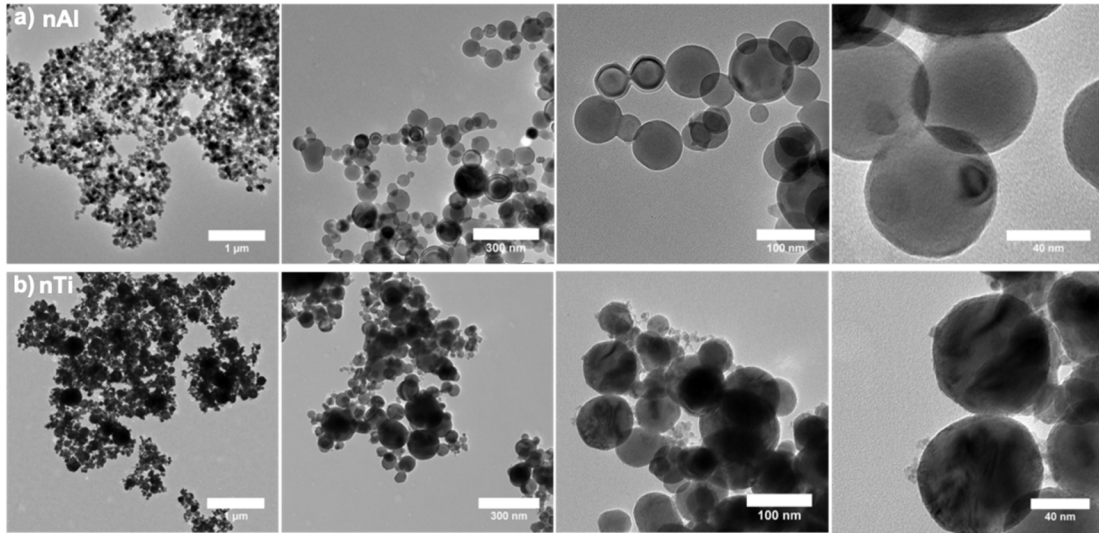
This work investigated the localized ignition of metal powders and metallized 3D printed propellants via microwaves. Materials were subjected to microwaves via a needle applicator which was able to ignite titanium nanopowders and energetic composites containing high mass loadings of Ti (~65%). Aluminum powders and Al/PVDF propellants were unable to be heated with the needle applicator after 30 seconds of exposure. Titanium’s observed interaction with microwave radiation is unexpected from the point of view models of bare metallic nanoparticle-microwave coupling with no consideration of the native oxide layer composition. These observations reveal the necessity of rigorous models which consider the properties and role of the oxide layer of energetic metallic nanoparticles in microwave heating. Despite superior microwave absorption of nanoscale titanium, the superior burn velocity and reduced cost of Al/PVDF makes it a more attractive propellant for traditional propulsion applications. The Ti/PVDF composite proved to be a suitable microwave-based igniter for solid propellants in both aerobic and inert environments. This work sets the stage for studies which focus on shell composition for enhanced microwave coupling of heterogeneous energetic systems as well as for highly tailorable

combustion of solid rocket propellants that can be remotely ignited and staged with directed microwave radiation.

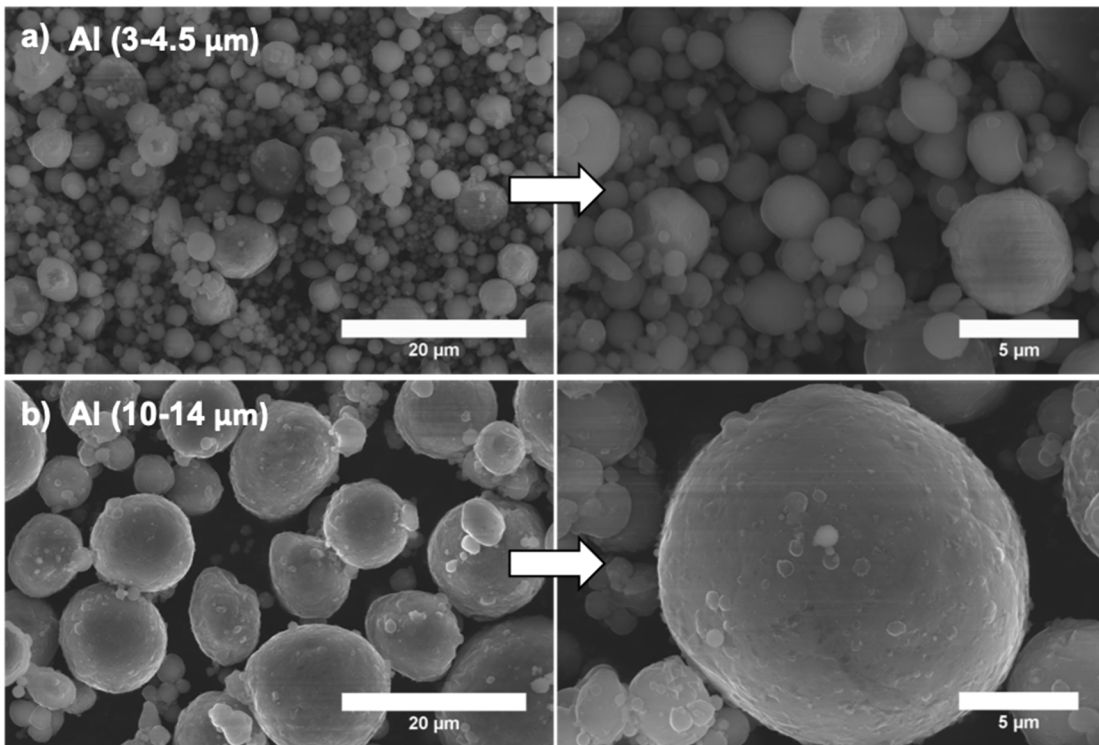
## ***9.6 Acknowledgements***

The authors gratefully acknowledge support from the AFOSR. Electron microscopy was performed on the NNS450 Scanning Electron Microscope in the Central Facility for Advanced Microscopy and Microanalysis (CFAMM) at University of California, Riverside. We would also like to acknowledge Dr. Brian Beaudoin at the University of Maryland Institute for Research in Electronics and Applied Physics (IREAP). Reproduced from D.J. Kline\*, M.C. Rehwoldt\*, C.J. Turner, P. Biswas, G.W. Mulholland, S.M. McDonnell, M.R. Zachariah, Spatially focused microwave ignition of metallized energetic materials, J. Appl. Phys. 127 (2020) 55901. doi:10.1063/1.5134089 with permission from AIP Publishing. \* Denotes that authors contributed equally to this work.

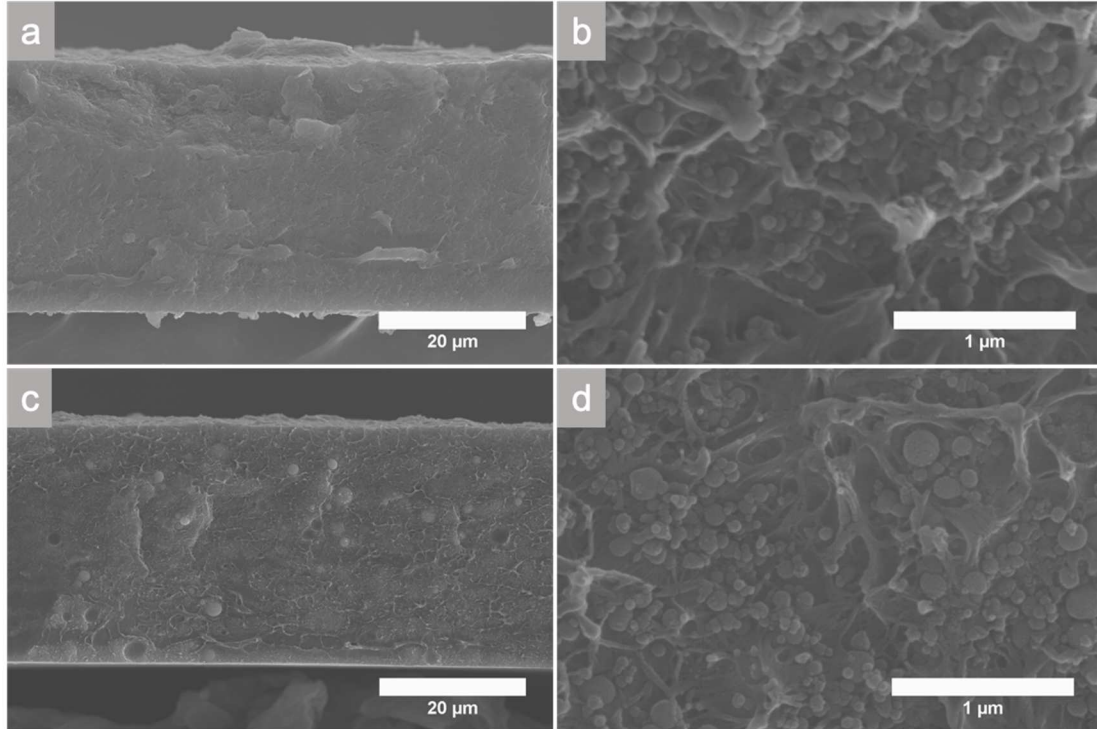
## 9.7 Supplemental Information



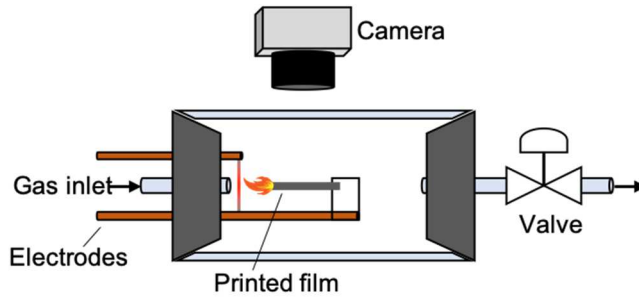
**Figure 9-7:** Transmission electron microscopy images of the as-received (a) nano-Aluminum (Novacentrix) and (b) nano-Titanium (US Research Materials). Images taken with a FEI Tecnai12.



**Figure 9-8:** Scanning electron microscopy images of the cross-section for as-received (a) 3-4.5 $\mu\text{m}$  diameter Al micron particles (Alfa Aesar) and (b) 10-14 $\mu\text{m}$  diameter Al micron particles (Alfa Aesar). Images taken with an FEI NNS450.



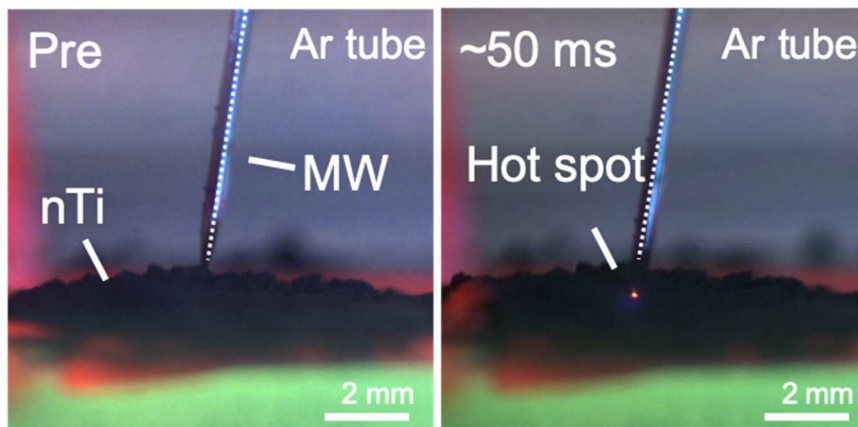
**Figure 9-9:** Scanning electron microscopy images of the cross-section for (a-b) Al/PVDF and (c-d) Ti/PVDF films. Each printed sample is 4 layers thick. Images taken with FEI NNS450.



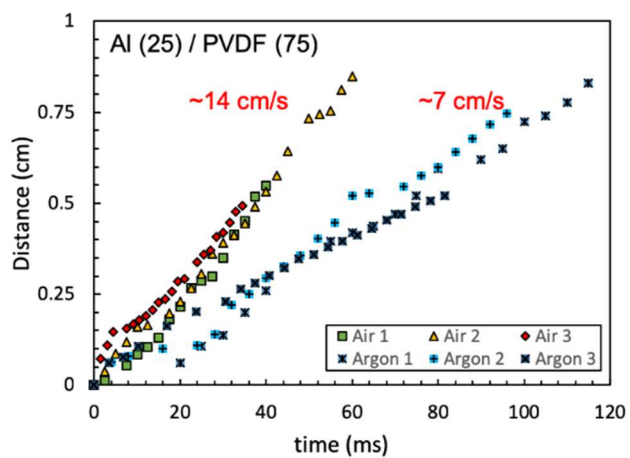
**Figure 9-10:** Macroscopic burn test apparatus schematic with gas inlets allow for environment control and ignition of a free-standing film using a resistively heated nichrome wire.

**Table 9-3:** Physical constants used for the complex dielectric constant and complex magnetic constants used in calculations to estimate the intensity fraction absorbed and the maximum temperature.

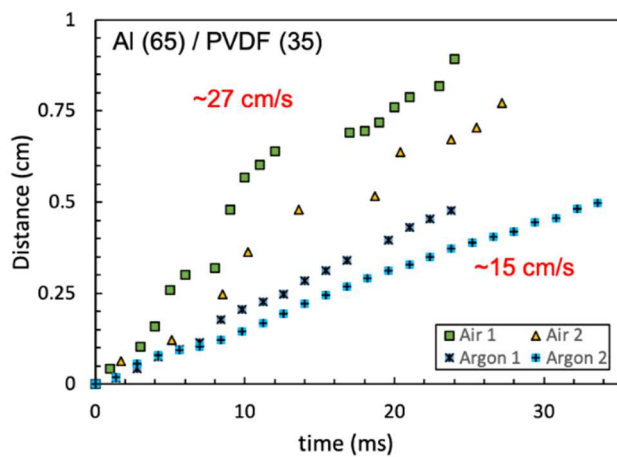
<i>Material</i>	$\tilde{\epsilon}_r'$	$\tilde{\epsilon}_r''$	$\mu'$	$\mu''$	<i>Source</i>
<i>Al</i>	0	$2.5 \times 10^8$	1	0	<sup>277</sup> , Azo Materials
<i>Al<sub>2</sub>O<sub>3</sub> (amorphous)</i>	9.8	$1 \times 10^6$	1	0	<sup>273</sup>
<i>Ti</i>	0	$1.3 \times 10^7$	1	0	<sup>277</sup> , Azo Materials
<i>TiO<sub>2</sub> (amorphous)</i>	33	10.4	1	0	<sup>289</sup>
<i>TiN</i>	15	4.2	1	0	<sup>288</sup>



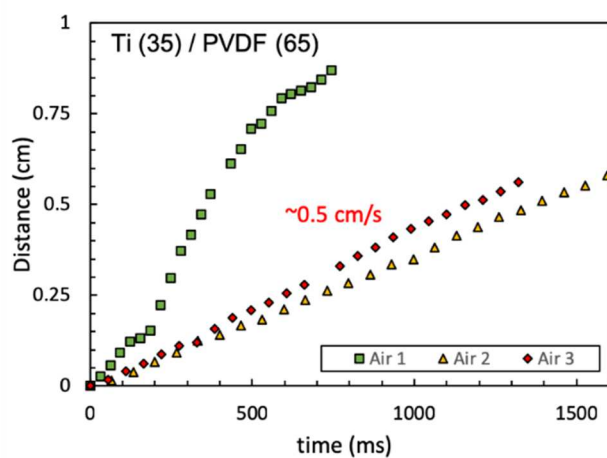
**Figure 9-11:** Images from microwave heating and ignition experiments for nano-titanium as performed in a glass tube with an inert Argon environment. Right image shows small heating point immediately around the microwave source.



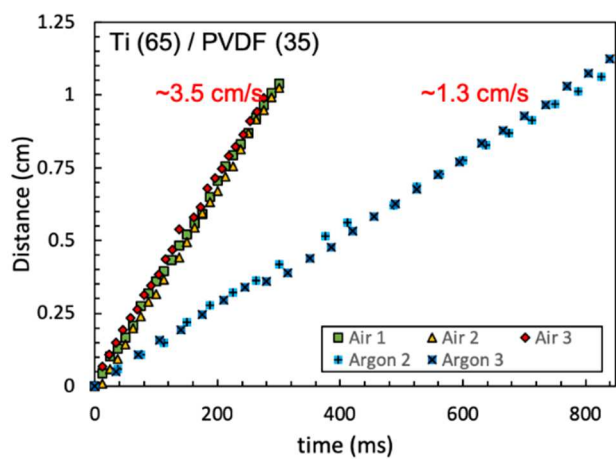
**Figure 9-12:** Distance vs. time data used to estimate propagation rate of a Al(25)/PVDF(75) mixture in air and argon.



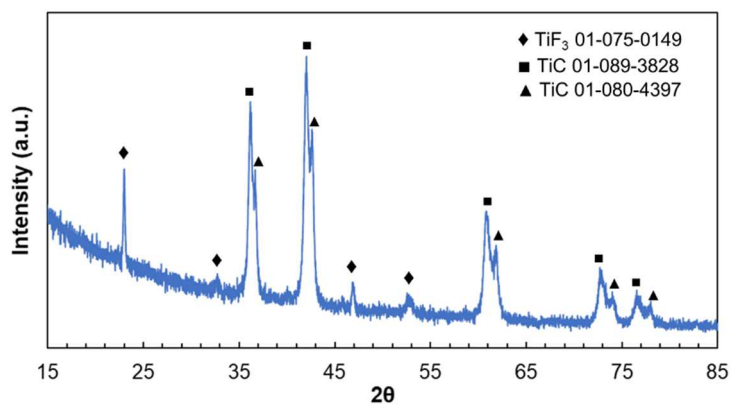
**Figure 9-13:** Distance vs. time data used to estimate propagation rate of a Al(65)/PVDF(35) mixture in air and argon.



**Figure 9-14:** Distance vs. time data used to estimate propagation rate of a Ti(35)/PVDF(65) mixture in air. This sample did not propagate in argon.



**Figure 9-15:** Distance vs. time data used to estimate propagation rate of a Ti(65)/PVDF(35) mixture in air and argon.



**Figure 9-16:** Ti-PVDF argon environment post combustion products.

## Chapter 10: Summary

### *10.1 Conclusions*

This dissertation has consisted of three generalized efforts to improve our understanding of structure function relationships in energetic materials: (1) development of new diagnostics and methods to probe reactions *in-operando*, (2) manipulation of reactivity by introducing variations in the reaction chemistry, and (3) identification of methods to vary combustion behavior by utilizing external forces or non-reactive materials. Many of the concepts are interwoven throughout the work presented, but the overarching conclusion of the studies remains consistent – careful manipulation of chemical and physical phenomena, whether introduced during material preparation or in reaction, enables similarly precise control over combustion behavior which can ultimately be integrated into making highly-specialized propellants with tunable reactivity.

This dissertation begins with the development of high-speed diagnostic methods which can capture spatial and temperature variations of combusting energetic materials (Chapter 2). These diagnostics were then applied to a specific case study which had only been previously explored without diagnostics that did not provide temperature information. In particular, a careful balance of the reaction mechanisms of Al-based nanothermites between high energy release ( $\text{WO}_3$ ) and gas release ( $\text{Fe}_2\text{O}_3$ ) produced a variation in pressurization rate, peak pressure, burn time, and temperature for hybrid composites. Furthermore, it was found that certain mixtures (Al/30  $\text{Fe}_2\text{O}_2$ /70  $\text{WO}_3$ ) even outperformed the pure compounds since gas release from the  $\text{Fe}_2\text{O}_3$



decomposition prevents reactive sintering between Al/WO<sub>3</sub> which enhances energy release in a feedback-loop style mechanism.

Considering that reactive sintering in energetic materials poses a large barrier to realizing potential energy release rates, a scalable nanomanufacturing method for high metal content aluminum nanoparticles was explored in Chapter 3. It was found that sub-5 nm aluminum particles could be generated by laser shock heating larger micron sized particles that were embedded in a reduced graphene oxide (rGO) matrix at metal loadings exceeding 50 wt%. Subsequently, the nanoparticle-laden reduced graphene oxide sheet could be ignited in an oxygen environment where it was found that the thermal shock time, and thus particle size, impacted the burn time of the sheets. However, since the particles were encapsulated in the rGO sheets, there were diminishing returns on the burn time for different levels of shock heating since there was significant levels reactive sintering between the tightly packed particles.

Understanding reaction dynamics on the particle scale are paramount to improving energy output and tuning reactivity of materials, though visualizing the reactions on the microscale has not been widely explored in energetic materials. In Chapter 4, a high-speed microscopy and thermometry apparatus was used to probe the reaction dynamics of a 90 wt% Al/CuO composite propellant burn stick *in-operando* with  $\mu$ s and  $\mu$ m resolution. This technique was used to observe and quantify the time and length scales of reactive sintering. Post-reaction forensics of the particles enabled one-for-one matching of particles in the captured videos and final products. Temperature measurements of the reaction front also allowed for heat flux estimations, which revealed that the propagation rate of the burning composite was dominated by

hot, reacting particles transferring heat to nearby unreacted areas by convection and radiation.

Having identified that heat transfer rates and reactive sintering can dramatically impact the propagation rate in energetic materials, it was desirable to minimize reactive sintering using a gas-generating oxidizer. In Chapter 5,  $\text{I}_2\text{O}_5$  was employed as a gas-generating oxidizer in high-loading composite burn sticks to reduce reactive sintering during reaction. While the  $\text{I}_2\text{O}_5$  did successfully reduce the final sintered particle size, samples did not consistently propagate. A simple heat transfer analysis revealed that the elimination of a condensing metal oxidizer significantly reduced the amount of energy that could be liberated for propagation. To overcome this heat transfer limitation,  $\text{CuO}$  was added to the composite so that the  $\text{Cu}$  vapor could participate in heat transfer and it was found that this significantly enhanced reliability in propagation without dramatic impacts to the degree of reactive sintering. Furthermore, the heat transfer rates of advecting particles was also quantified using the high-speed microscopy and thermometry method previously explored in Chapter 4.

While reductions in reactive sintering are important for gas-generating energetic materials, it offers only limited insight to the condensed phase heat transfer mechanisms in reacting materials. In Chapter 6, attention shifts to condensed-phase reactions of  $\text{Al/Zr/C}$  to probe the role of thermophysical properties on propagation. Using the high-speed thermometry apparatus, temperature profiles were used to estimate the thermal diffusivity of materials *in-operando* and estimate kinetic parameters (such as activation energy) for the reactions.

High-speed microscopy images captured in Chapter 4 revealed that the front is not planar as it appears in macroscopic videos but is instead very corrugated. Since the area of the burning surface should scale directly with the energy release/mass consumption rate of the reaction, by changing the microstructure of the reaction front, one could theoretically change the propagation rate using inert additives which change heat transfer rates in the material. This concept is explored in Chapter 7 where the high-speed microscopy apparatus was used to visualize the reaction front of 3D printed Al/PVDF films which had small additions of thermally insulating ( $\text{SiO}_2$ ) or thermally conductive (graphite) homogeneously dispersed in the films. It was found that the thermally insulating additives prompted inhomogeneous heat transfer in the reaction front which increased the corrugation and thus the propagation rate of the propellant. This counterintuitive result offers unique insight into a method that could be readily adapted into modern propellant manufacturing techniques using only small weight additions ( $\sim 2.5$  wt%).

The counterintuitive result in Chapter 7 subsequently led to a unique question on the role of additive geometry on heat transfer and energy release rates in propellants. In Chapter 8, numerical simulations of a surrogate reacting system with additives of different aspect ratio and thermal diffusivity homogeneously dispersed revealed a unique phenomenon where high-aspect ratio additives at low weight percent additions actually increased the propagation and energy release rates. This demonstrated the potential effect of thermal percolation on energetic material combustion where randomly oriented wires can actually lead to a conductive network of heat transfer highways where energy liberated by the reaction can pre-heat unreacted regions and

lower the energy barrier necessary to overcome for ignition. This work sets the foundation for another readily adaptable technique that could enable highly tunable propellant combustion behaviors with enhanced efficiency.

All of the aforementioned methods to tune the reactivity of an energetic material are inherently limited in versatility since they only offer versatility up the manufacturing stage. Modern solid rocket technology still does not offer a way to throttle a solid propellant *in-operando* which is a significant drawback in comparison to liquid propulsion. In Chapter 9, a method to ignite energetic materials remotely was explored where it was found that titanium nanoparticles were particularly susceptible to microwave-frequency oscillations in the electromagnetic field and would ignite at relatively low powers. However, Al powders would not ignite under any power. Further investigation and simulations on the microwave absorption properties of these particles found that the core-shell structure of the particles plays a significant role in the susceptibility of the materials to microwave radiation. In the case of Ti nanoparticles, the unique  $\text{TiO}_2/\text{TiN}$  shell composition actually enhanced the microwave absorption properties of the particles. Unfortunately, though, Ti/PVDF is a slow-propagating propellant that does not sustain combustion in inert environments. To overcome this, a composite structure consisting of Al/PVDF and with a Ti/PVDF sensitized region overlaid was prepared and demonstrated a useable propellant that could be remotely ignited. This work set the stage for remotely ignited, staged, and throttled propellants that could be manipulated with directed energy and offer an unprecedented level of control over energy release in solid propellants.

## ***10.2 Recommendations for future work***

While this dissertation has explored and demonstrated some avenues in which a reactive material could be prepared with tunable reactivity, it has perhaps raised as many questions as it has answered. Two projects are proposed below for follow up efforts to the research discussed in this dissertation, both which focus on a hybrid project involving calculation and experimentation.

### ***10.2.1 Exploiting additive geometry for enhanced heat transfer in energetic materials***

As has been discussed throughout the entire body of work presented, the propagation rate of a reacting material scales with the chemical reaction rate and the thermal diffusivity of the propagating medium. In Chapter 7, the role of additives with varied thermal diffusivity on the propagation rate of a propellant was examined using high speed microscopy techniques that revealed a counterintuitive result – adding a material which has a lower thermal diffusivity increased the propagation rate since the degree of corrugation in the reaction front was increased.<sup>174</sup> However, this posed a significant question regarding the studies that demonstrated how large, thermally conductive rods can also increase propagation rate in burning propellants. It is proposed that the geometry of the conductive additives in this dissertation's study (planar flakes) may actually be a detriment to the propagation rate since the energy is not driven along a linear path like would be done with the addition of long, conductive rods. The addition of large masses is not ideal for scalable additive manufacturing techniques, though, so it was proposed that alternative geometries could be explored such as nanowires with high aspect ratios. When adding smaller nanowires with high aspect ratios, there is an increasing statistical chance that a conductive pathway will be formed by the

connection of multiple conductive nanowires which can result in a phenomenon called thermal percolation. This proposal gave rise to the study in Chapter 8 which estimated energy release with these high aspect ratio additives and proposed that low volume percent additions of high aspect ratio materials could enhance energy release per unit volume.

While the study in Chapter 8 does reveal an interesting paradigm which has not been explored and would lend itself to an important experimental follow up study for the energetic and heat transfer communities, early experimentations of this phenomena were stopped in favor of exploring the numerically simulated aspect of this project first for feasibility. Having demonstrated that this avenue is feasible numerically, identifying materials which could reflect this phenomenon is difficult. These experiments would need to identify a high-aspect ratio additive which is relatively inert and has a high thermal diffusivity – the obvious choice here would be carbon nanowires or carbon nanofibers. To be relatively comparable to the simulations, the material would also likely need to propagate at relatively slower rates and would not generate significant amounts of gas. Either a gasless reacting material (like ball milled powders in Chapter 6) or high loading burn sticks would be ideal candidates. It is unlikely that far-reaching heating of the additives would be observable using the high-speed camera in the visible wavelengths, so it would be desirable to have access to a high-speed infrared camera which can identify temperatures over a broad range. Experimental methods described in other chapters included in the dissertation would also be of particular use to probe the reaction front corrugation. This study would be an experimental complement to the simulations discussed in Chapter 8 and could serve as

a platform for future studies that wish to vary the aspect ratio and composition of the additives.

#### 10.2.2 Microwave preheating as a method to throttle a solid propellant in-operando

Similar to the experiment proposed in section 10.2.1 where additives may serve as a means to channel energy in regions far away from the reaction front to tailor energy release, an in-operando method of changing energy release for energetic materials could be explored by incorporating localized heating methods utilized in Chapter 9. Whereas additives essentially act as pre-programmed zones that would be decided during manufacturing of a propellant, by making the segments of a reactive material sensitive to external energy inputs like microwave radiation, directed energy could lead to preheating that effectively lowers the energy barrier that must be crossed to sustain propagation. Modern solid rocket motors lack acceleration or deceleration capabilities in-operando, but demonstration of a working method to throttle these materials could give rise to a new class of propellants which could be easily adapted into modern configurations and provide a new versatility to an old technology.

Testing of these materials would require a more rigorous characterization of the microwave generating apparatus and a high-speed infrared camera to identify the degree of preheating during the combustion event. A high-speed camera (either visible or infrared) could be used to observe a precisely timed combustion event which originates from a point offset from the sphere of influence by the microwave source. The propagation would then cross through this sphere of influence where either the microwave source was powered on an actively radiating or was disconnected and the propagation rate through that region would be tracked afterwards using the data

retrieved from the video. Considering that gases released during propellant reactions may interact with the microwave and cause undesirable side effects, condensed phase reacting materials would be ideal candidates for this study (like those discussed in Chapter 6). Furthermore, since the focus of this study would be on heating rather than ignition, any material could be used in principle so long as the temperature of the heated region could be readily measured with a camera. Key parameters which should be varied are the power input to the material being heated and analysis could focus on the degree to which the energy barrier is lowered and its relation to propagation rate through the preheated region. The results of this work could also be coupled into propagation simulations to design complex architectures with predictable performance for a variety of applications.



## Appendices

### Appendix A: Experimental description and useful information for color camera pyrometry

#### *A.1. Introduction*

The color camera pyrometry apparatus has been discussed in detail in other articles, notably in Jacob & Kline et al.,<sup>73</sup> and was designed to estimate the temperature of a combusting material with spatial and temporal resolution. It has been modified heavily over the years since it was first started as part of my research. The following appendices will include more detailed descriptions of how to calibrate the camera and will show the raw MATLAB code for the software.

#### *A.2. Descriptions of available cameras*

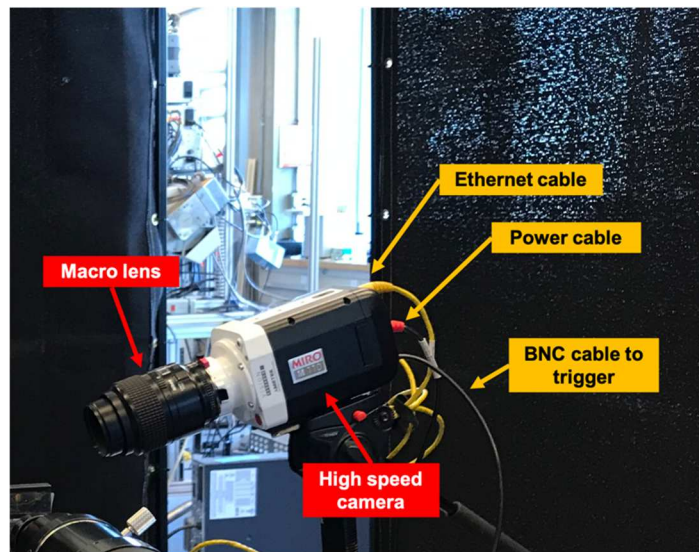
The cameras that have been used throughout the course of the research above are summarized in Table A-1 with detailed descriptions on the make and pyrometry capabilities. The sensors on the Vision Research Miro M110 is the same as the one utilized in the Vision Research VEO710L, therefore the online documentation for the cameras will show that the spectral response of the cameras is the same. As a result of this, the cameras can be calibrated simultaneously for the same lens.

**Table A-1:** Cameras available for use in the Zachariah Group as of October 2020.

<b>Manufacturer</b>	<b>Model</b>	<b>Max. Framerate</b>	<b>Memory</b>	<b>Pyrometry</b>
Vision Research	V12.1	1,000,000 @ 128x8	8GB	No
Vision Research	Miro M110	400,000 @ 64x8	10GB	Yes
Vision Research	VEO710L	700,000 @ 64x8	20GB	Yes

### ***A.3. Procedure for performing color camera pyrometry experiments***

To perform color camera pyrometry, it is recommended to trigger a heating wire (or other input power source) simultaneously with the camera triggering (discussed below) using a standardized TTL signal from any of the available trigger boxes in the lab (see Appendix E or Appendix F for sample TTL output boxes). A typical setup of a color camera pyrometry experiment can be seen in Figure A-1, but this is only a common example that is used for tabletop measurements. A simple procedure of setting up the color camera is provided below. Detailed use of the Phantom Camera Control software is left to the user since documentation by Vision Research is readily available and provides good detail on operation.



**Figure A-1:** Example camera setup for Vision Research Phantom Miro M110 equipped with a Nikon macro lens used for pyrometry. Trigger input should be a TTL signal (5V).

1. Set up the camera on a stabilized surface (e.g. optical table, tripod) with the desired lens and connect to power and ethernet.
2. On a windows computer, start the Phantom Camera Control (PCC) software and connect to the camera.

\*If the camera does not automatically connect, close PCC, change the network settings on the computer and set the IP address to 100.100.100.1 and the subnet mask to 255.0.0.0, then restart PCC.

3. If using a TTL triggering option on the camera, connect the Trigger input on the camera to the signaling device that will send the TTL signal.
4. Click “Low light” in PCC and focus on the desired object manually using the lens,  $f\#$ , and window cropping options in PCC. Once focused, disable “Low light” option.
5. Set the desired framerate and exposure for the video to be taken.
6. Click CSR to save a closed shutter reference at the recording settings.
7. Click “Capture” in PCC.
8. Run experiment.
9. Review video in PCC by cropping the length of the video.
10. Save .cine file, make sure that the check box next to 16 bpp is enabled.
11. Repeat steps 4-10 for each experiment to be run.
12. Process videos for temperature measurements using the pyrometry codes as detailed in the software manual.

#### ***A.4. Recommendations for performing pyrometry experiments***

Taking good high-speed videos and getting the right settings for reliable color camera pyrometry results may require a dissertation itself, but some useful information for getting started is provide below and has been disseminated to users of the experiment when they first started. Truly, the only way to get really good at this experiment is to practice and prepare to fail (a lot).

The most important aspects of high-speed camera operation for good pyrometry results is consideration of the brightness of the samples being tested. In particular, videos should avoid saturating the camera sensors (since temperature cannot be measured there) but should not be too dark that there is a poor signal-to-noise ratio. Changing brightness is generally limited by the exposure time for each frame and the aperture size ( $f\#$ ) on the lens. A table of suggested settings can be seen in Table A-2 for a variety of experiments, but it is best to prepare enough sample for testing of at least 2-3 practice runs.

**Table A-2:** Recommended settings for a variety of experiments as a starting point for color camera pyrometry operation.

Experiment type	Framerate	Exposure	$f\#$
Carbon heat shock (CNF), 10ms	41000 fps	60 $\mu$ s	2.8
Carbon heat shock (CNF), 55ms	10000 fps	10 $\mu$ s	2.8
Carbon heat shock (CNF), 505ms	3000 fps	4 $\mu$ s	2.8
RGO Laser Shock, 250ms+	3000 fps	330 $\mu$ s	2.8
RGO Laser Combustion	3000 fps	72 $\mu$ s	32
Microwave heating in oven	5000 fps	20 $\mu$ s	32
Thermite wire (Al/Fe <sub>2</sub> O <sub>3</sub> /WO <sub>3</sub> )	20000 fps	4 $\mu$ s	22

Some other useful tips for color camera pyrometry are below:

- Make sure that there is enough extra material to account for 2-3 test runs of pyrometry to determine settings for optimal results.
- Before calculating temperatures, check videos by looking at the RGB values of the bright frames with a gain of 1 in PCC. Ideal values for R should be ~150-200, but none of the regions should be >225.
- As of October 2020, videos >6GB can generally not be processed on a computer with 16GB of RAM using version 10 of the color camera pyrometry software. This is a limitation in the way that MATLAB calculates the new matrices, which requires

a duplication of the matrix in the memory and would thus require at least 12GB of free memory. In many modern computers at this time, this is unlikely to be the case.

- Try reducing exposure before changing  $f\#$ . Neither will change the temperature measurement results, though it is much easier to forget changing an  $f\#$  before running the experiment. This being said, if you feel that your videos are far too dark, it might be useful to try increasing the  $f\#$  first since the amount of light increases exponentially with changes in  $f\#$  and linearly with changes in exposure.
- “Hidden” calibration curves and options for filtering are in the software package and detailed in the software manual. At the prompt for “Lambda Correction”:
  - ‘n’ – Ideal emitting case. Best used for heating of carbon materials.
  - ‘y’ – Thin film front calculation which makes emissivity a function of wavelength. Best used for combusting thermite systems.
  - ‘t’ – Extends the temperature of the calibration curve from 773-5773K. Useful for very high temperature chemistry but should be used with caution.
  - ‘h’ – Widens the bounds for ratio thresholding. Useful in situations where there may be a lot of odd black (or gray) spots.
- The equations below can be used to predict or determine how a video should be recorded if videos are becoming too large to process or save.

$$\frac{\text{frames}}{\text{event}} = \text{event length (ms)} * \frac{1}{1000} * \text{framerate (fps)}$$

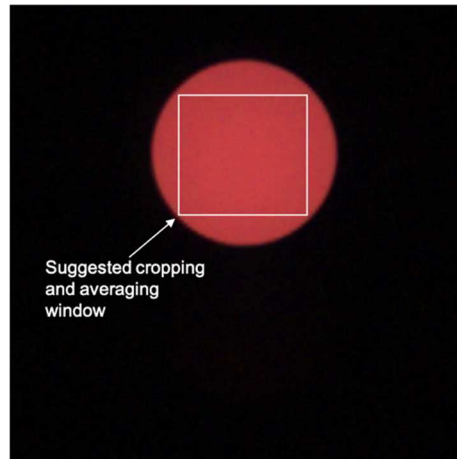
$$\text{video size} = \frac{\text{event length (ms)}}{\text{max.rec.time (s)}} * \frac{1}{1000} * \text{\#events} * \text{memory (GB)}$$

## Appendix B: Calibrating a color camera pyrometer

### *B.1. Procedure for calibrating camera pyrometer*

Calibration of a color camera pyrometer should be done by taking videos of a blackbody source consisting of at least 100 frames and using the exact optics that are expected to be used in the experiments. Exposure times for the videos need to be chosen by the operator at the time of taking the calibration videos, but they should follow the same principles of exposure described in Appendix A – videos should be neither saturated nor dark. A simple procedure is described below.

1. Turn on the camera and connect to the camera using a computer running the corresponding camera control software.
2. Turn on blackbody source to ~500 C (as a starting point) and allow to stabilize at that temperature.
3. Set up the camera with the optics in front of the outlet of the blackbody source at ~1 focal length away or further. Use a knife edge to focus on the outlet of the blackbody.
4. Adjust camera settings to focus on the area around the blackbody outlet and take a closed shutter reference (CSR).
5. Take a video of at least 100 frames of the blackbody (more is preferable) and record the temperature and recording settings. An example image of a blackbody can be seen in Figure B-1.



**Figure B-1:** Single image from a color camera calibration video as a demonstration. Red circle is outlet of Newport Oriel blackbody source.

6. Save the video and record temperature at time of recording.
7. Increase the temperature on the blackbody, allow to stabilize, and repeat steps 5-6 in desired temperature increments over the desired temperature window. Generally, going up to the maximum temperature on the blackbody will result in the most calibration ratios.
8. Using MATLAB (or some other computational program), import the raw data into the workspace and demosaic the image using the corresponding color filter on the camera. For Vision Research Phantom cameras, the filter is a Bayer filter with GBRG patterning.
9. Average the value of the individual color channels (typically red, green, blue) over the blackbody region over the duration of the video. Save values and corresponding temperature.
10. Calculate the ratios in the different channels (e.g. R/G, R/B, G/R, G/B, B/R, B/G) using the averaged values measured in step 9.

11. Calculate the expected channel ratios using the equations below where  $I_i$  is intensity of a channel,  $\chi_i$  is the spectral response of the channel as a function of wavelength,  $\psi_i$  is channel gain, and the Radiance ( $L$ ) is calculated from Planck's Law.

$$\frac{I_i}{I_j} = \frac{\psi_i \int L(\lambda, T) \chi_i(\lambda) d\lambda}{\psi_j \int L(\lambda, T) \chi_j(\lambda) d\lambda}$$

12. Calibration ratios are calculated using the equation below where  $C_{ij}$  is the calibration ratio where the values are nominally constant over the entire temperature window under consideration.

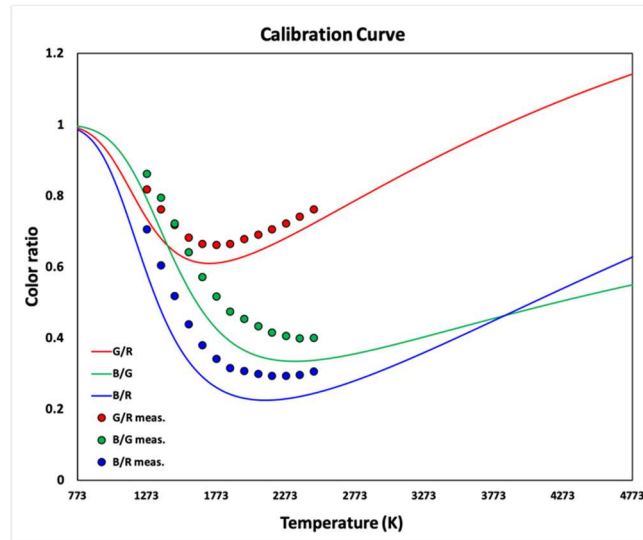
$$\left(\frac{I_i}{I_j}\right)_{exp} = C_{ij} \left(\frac{I_i}{I_j}\right)_{theor}$$

A typical calibration curve calculation spreadsheet can be seen in Figure B-2 with the corresponding calibration curve plots shown in Figure B-3.

Temperature (K)	Measured									Factor						
	R mean	G mean	B mean	G/R meas.	B/G meas.	B/R meas.	R/G meas.	G/B meas.	R/B meas.	G/R	B/G	B/R	R/G	G/B	R/B	
1273	631	516	445	0.8177496	0.8624031	0.70522979	1.22286822	1.15955056	1.41797753	1.11152523	1.09008914	1.21166158	0.89966469	0.91735617	0.82531295	
1373	1023	779	619	0.76148583	0.79460847	0.60508309	1.31322208	1.25848142	1.65266559	1.11844242	1.12916695	1.26290822	0.89410056	0.88560863	0.79182318	
1473	1082	777	561	0.7181146	0.72200772	0.51848429	1.39253539	1.38502674	1.92869875	1.11736556	1.17026993	1.30761931	0.89496225	0.85450372	0.76474857	
1573	1136	776	498	0.68309859	0.64175258	0.43838028	1.46391753	1.55823293	2.2811245	1.10135836	1.18952639	1.31009483	0.90796968	0.84067072	0.76330352	
1673	1161	772	441	0.66494401	0.57124352	0.37984496	1.50388601	1.70506689	2.63265306	1.08917433	1.20059915	1.30766178	0.91812667	0.83291747	0.76472374	
1773	1167	772	399	0.66152528	0.51683938	0.34190231	1.51165803	1.93483709	2.92481203	1.08388423	1.21147598	1.31309971	0.92260776	0.82543939	0.76155679	
1873	1153	766	364	0.66435386	0.47519582	0.31569818	1.50522193	2.1043956	3.16758242	1.07663978	1.21661828	1.30985963	0.92881577	0.8219505	0.76344058	
1973	1136	770	350	0.6778169	0.45454545	0.30809859	1.47532468	2.2	3.24571429	1.07813096	1.24361896	1.34078411	0.9275311	0.80410482	0.74583223	
2073	1114	769	333	0.69030521	0.43302991	0.2989228	1.44863459	2.30930931	3.34534535	1.07228926	1.24035949	1.33002416	0.93258418	0.80621788	0.75186604	
2173	1095	773	321	0.70593607	0.4152652	0.29315068	1.41655886	2.40809969	3.41121495	1.06761467	1.22327802	1.30598956	0.93666753	0.81747566	0.76570291	
2273	1064	769	312	0.72274436	0.40572172	0.29323308	1.38361508	2.46474359	3.41025641	1.06235468	1.21121236	1.28673712	0.94130521	0.82561905	0.77715952	
2373	1043	774	309	0.74209012	0.39922481	0.29626079	1.34754522	2.50485437	3.37540453	1.05935411	1.19391349	1.26477716	0.94397141	0.83758163	0.79065311	
2473	994	758	304	0.76257545	0.40105541	0.30583501	1.31134565	2.49342105	3.26973684	1.0570794	1.1909467	1.25892523	0.94600273	0.83966814	0.79432835	
RATIOS											1.0842	1.1932	1.2931	0.9226392	0.8391626	0.77388088
STDEV											0.02090157	0.04133291	0.03368429	0.01765256	0.03038672	0.02067257

**Figure B-2:** Single image from a color camera calibration video as a demonstration. Red circle is outlet of Newport Oriel blackbody source.





**Figure B-3:** Example calibration curve for Phantom VEO710L equipped with an Infinity Optics K2 Distamax Lens, an NTX tube, and a CF-4 objective. Blackbody source was a Mikron M390. Measurements were performed at the Naval Air Weapons Station at China Lake (NAWS-CL) in August 2019.

### ***B.2. Program for extracting single channel values for calibration ratio calculations***

For the case of using a Vision Research Phantom Camera, the averaging and determining of the RGB values can be performed using the MATLAB code below. The MatCine software (see Appendix C) is required for use of this software. Make sure to crop the window down to just the area when the option is presented.

```
%% Channel Information Extraction for Pyrometry Calibration
% Dylan J. Kline
% Chemical Engineering - Zachariah Research Group
% University of Maryland College Park

clc
clear
close all

% Adds all external functions to working path.
addpath(genpath(fileparts(which(mfilename))))

%% 1. Import of Images

% Pick .cine or .mat file using ui - multiple files can be selected
[files, path]=uigetfile({'*.cine;*.mat'}, 'Pick a Cine File'); % GUI
selection of files
filename=fullfile(path, char(files)); %
Extracts the filename and path
[pathstr,name,ext]=fileparts(filename); %
separates parts of filename and path
```

```

% Extraction of images using pycine (before) or reruns
if ext(end-3:end)==' .mat'
    load(filename, 'raw_image_array') %
should import the data as a matlab file
    %raw_image_array=uint16(raw_image_array); %
converts to uint16 for faster processing
else
    [header, bitmap, setup, raw_image_array]=MatCine(filename); %
runs MatCine program to extract information
end

%% Averaging R/G/B Over Entire Window and Video

raw_mean = mean(raw_image_array,3); % average values for RGB in
the time dimension
raw_mean=uint16(raw_mean); % convert values to 16 bit
integers
demosaicIm = demosaic(raw_mean, 'gbgr'); % demosaic the image using
Bayer filter pattern
R = double(demosaicIm(:, :, 1)); % Separating Red, Green, and
Blue color layers
G = double(demosaicIm(:, :, 2));
B = double(demosaicIm(:, :, 3));

Rmean = uint16(mean(mean(R))); % Average of Red channel
Gmean = uint16(mean(mean(G))); % Average of Green channel
Bmean = uint16(mean(mean(B))); % Average of Blue channel

tosheet = [Rmean Gmean Bmean]; % Option to make copying to
spreadsheet easier
disp(tosheet)

```

### ***B.3. Program for calculating predicted ratios for color camera pyrometer***

The program below can be used to calculating the predicted channel ratios and the ratios\_summary variable can be saved for import into the color camera pyrometry code. A file is required for importing the spectral response curve of the camera as a function of wavelength. The format of the spectral response curve should be in the format of [wavelength (m), R response curve, G response curve, B response curve].

```

%% Theoretical RGB Ratio Calculations for Miro M110 Camera
% Dylan Kline
% Chemical Engineering
% University of Maryland College Park - Zachariah Group

% The following code is designed calculate spectral response
integrals for

```

```

% blackbodies on a Phantom Vision Miro M110 Camera. Spectral
response
% curves have been pulled out through binarization in a separate
file.

%% Reset & Import
%clc
%clear
%close all

load('phil_spectral_response_wbump.mat')
load('beamsplit_info.mat')
load('microscope_info.mat')
% Wavelength (m) | R Response Curve | G Response Curve | B Response
Curve |
% Please note, in the image documenting the response of the camera,
the RGB
% curves overlay and a "bump" is present in the far infrared region.
This
% has been taken into account.

%% Blackbody Curve
% Plancks Law for a Blackbody Object
C1=37413*10^(-20); % (W/m2)
C2=14338*10^(-6); % (mK)
epsilon = @(lambda) 1/lambda; % Assuming black body /lambda for
lambda corrections
L= @(lambda,T) C1*epsilon(lambda)/(lambda^5*(exp(C2/(T*lambda))-1));

%% Temperature Range (K)
T=773:5:4773;

%% Microscope Assembly
microscope = input('Microscope assembly w/ beam splitter? (0=no,
1=yes): ');

%% Red spectral response - Not integrated

k=1;
for i=1:size(phil_spectral_response,1)
    if isnan(phil_spectral_response(i,2))==0 % Check to see
there is a value for the response
        r_wl(k,1)=i; % Save the row of
the spectral response from the original file (book keeping)
        r_wl(k,2)=phil_spectral_response(i,1); % Get wavelength
at that index
        r_wl(k,3)=phil_spectral_response(i,2); % Get spectral
response at that wavelength
        k=k+1;
    end
end

if microscope==1
    for i=1:size(T,2) % Iterative
calculations for different temperatures
        for j=1:size(r_wl,1)

r_L(j,i)=L(r_wl(j,2),T(i))*r_wl(j,3)*beamsplit_fit(r_wl(j,2))*microsc
ope_fit(r_wl(j,2));
            end
        end
end

```

```

else
    for i=1:size(T,2)
        calculations for different temperatures
        for j=1:size(r_wl,1)
            r_L(j,i)=L(r_wl(j,2),T(i))*r_wl(j,3);
        interior of integral
        end
    end
end

%% Green spectral response - Not integrated
% Same process as for r
k=1;
for i=1:size(phil_spectral_response,1)
    if isnan(phil_spectral_response(i,3))==0
        g_wl(k,1)=i;
        g_wl(k,2)=phil_spectral_response(i,1);
        g_wl(k,3)=phil_spectral_response(i,3);
        k=k+1;
    end
end

if microscope==1
    for i=1:size(T,2)
        calculations for different temperatures
        for j=1:size(g_wl,1)
            g_L(j,i)=L(g_wl(j,2),T(i))*g_wl(j,3)*beamsplit_fit(g_wl(j,2))*microsc
ope_fit(g_wl(j,2));
        end
    end
else
    for i=1:size(T,2)
        calculations for different temperatures
        for j=1:size(g_wl,1)
            g_L(j,i)=L(g_wl(j,2),T(i))*g_wl(j,3);
        interior of integral
        end
    end
end

%% Blue spectral response - Not integrated
% Same process as for r
k=1;
for i=1:size(phil_spectral_response,1)
    if isnan(phil_spectral_response(i,4))==0
        b_wl(k,1)=i;
        b_wl(k,2)=phil_spectral_response(i,1);
        b_wl(k,3)=phil_spectral_response(i,4);
        k=k+1;
    end
end

if microscope==1
    for i=1:size(T,2)
        calculations for different temperatures
        for j=1:size(b_wl,1)
            b_L(j,i)=L(b_wl(j,2),T(i))*b_wl(j,3)*beamsplit_fit(b_wl(j,2))*microsc
ope_fit(b_wl(j,2));
        end
    end
else

```

```

        for i=1:size(T,2)                                % Iterative
calculations for different temperatures
            for j=1:size(b_wl,1)
                b_L(j,i)=L(b_wl(j,2),T(i))*b_wl(j,3);    % Calculate
interior of integral
            end
        end
    end

%% Trapezoidal Integration
% Performs trapezoidal integration over the entire spectral region.
for i=1:size(T,2)
    r_int(i)=trapz(r_wl(:,2),r_L(:,i));
    g_int(i)=trapz(g_wl(:,2),g_L(:,i));
    b_int(i)=trapz(b_wl(:,2),b_L(:,i));
end

%% Ratios & Transpositions
g_r=(g_int./r_int)';                                     % calculates integrated G/R values
for different temperatures
b_g=(b_int./g_int)';                                     % calculates integrated B/G values
for different temperatures
b_r=(b_int./r_int)';                                     % calculates integrated B/R values
for different temperatures
r_g=(r_int./g_int)';                                     % calculates integrated R/G values
for different temperatures
g_b=(g_int./b_int)';                                     % calculates integrated G/B values
for different temperatures
r_b=(r_int./b_int)';                                     % calculates integrated R/B values
for different temperatures

% Creates summary ratio matrix for transport to Excel
ratios(:,1) = T';
ratios(:,2) = g_r;
ratios(:,3) = b_g;
ratios(:,4) = b_r;
ratios(:,5) = r_g;
ratios(:,6) = g_b;
ratios(:,7) = r_b;

ratios_summary = ratios; % variable to save for import into camera
pyrometry code

% % Creates Polynomial Fits
% for i=1:size(ratios,2)-1
%     polyratios(:,i)=polyfit(ratios(:,1),ratios(:,i+1),6)';
% end

%% Ratio Figure
figure
subplot(1,3,1)
plot(g_r,T,'r','LineWidth',2)
ylabel('Temperature (K)','FontSize',18,'FontName','Times New Roman')
xlabel('G/R','FontSize',18,'FontName','Times New Roman')
title('Temperature (K) vs. G/R','FontSize',18,'FontName','Times New
Roman')
grid on

subplot(1,3,2)

```

```

plot(b_g,T,'g','LineWidth',2)
ylabel('Temperature (K)','FontSize',18,'FontName','Times New Roman')
xlabel('B/R','FontSize',18,'FontName','Times New Roman')
title('Temperature (K) vs. B/G','FontSize',18,'FontName','Times New Roman')
grid on

subplot(1,3,3)
plot(b_r,T,'b','LineWidth',2)
ylabel('Temperature (K)','FontSize',18,'FontName','Times New Roman')
xlabel('B/R','FontSize',18,'FontName','Times New Roman')
title('Temperature (K) vs. B/R','FontSize',18,'FontName','Times New Roman')
grid on

figure
hold on
grid on
plot(g_r,T,'r','LineWidth',2)
plot(b_g,T,'g','LineWidth',2)
plot(b_r,T,'b','LineWidth',2)
legend('G/R','B/G','B/R','Location','Southeast')
ylabel('Temperature (K)','FontSize',18,'FontName','Times New Roman')
xlabel('I_{ij} (Theoretical)','FontSize',18,'FontName','Times New Roman')
title('Temperature (K) vs. I_{ij} (Theoretical)','FontSize',18,'FontName','Times New Roman')
hold off

%% Spectral Response Plot
figure
hold on
area(1E9*r_wl(:,2),r_wl(:,3),'LineWidth',1.25,'EdgeColor','r','FaceColor','r','FaceAlpha',0.15)
area(1E9*g_wl(:,2),g_wl(:,3),'LineWidth',1.25,'EdgeColor','g','FaceColor','g','FaceAlpha',0.15)
area(1E9*b_wl(:,2),b_wl(:,3),'LineWidth',1.25,'EdgeColor','b','FaceColor','b','FaceAlpha',0.15)
xlim([1E9*min(phil_spectral_response(:,1)),1E9*max(phil_spectral_response(:,1))])
xlabel('Wavelength (nm)','FontSize',18,'FontName','Times New Roman')
ylabel('Spectral Response','FontSize',18,'FontName','Times New Roman')
legend('R','G','B')
title('Spectral Response Curve - Phantom Vision M110','FontSize',20,'FontName','Times New Roman')

%% Emissions Tests
figure
set(gcf,'Color','w')
hold on
yyaxis left
area(1E9*r_wl(:,2),r_wl(:,3),'LineWidth',1.25,'EdgeColor','r','FaceColor','r','FaceAlpha',0.15)
area(1E9*g_wl(:,2),g_wl(:,3),'LineWidth',1.25,'EdgeColor','g','FaceColor','g','FaceAlpha',0.15)
area(1E9*b_wl(:,2),b_wl(:,3),'LineWidth',1.25,'EdgeColor','b','FaceColor','b','FaceAlpha',0.15)
xlim([1E9*min(phil_spectral_response(:,1)),1E9*max(phil_spectral_response(:,1))])
xlabel('Wavelength (nm)','FontSize',18,'FontName','Times New Roman')

```

```

ylabel('Spectral Response','FontSize',18,'FontName','Times New
Roman')

yyaxis right
bar(aluminumemission(:,1)/10,aluminumemission(:,2),'k')
bar(copperemission(:,1),copperemission(:,2),'b')
bar(sodiumemission(:,1)/10,sodiumemission(:,2),'r')
ylabel('Normalized Emission
Intensity','FontSize',18,'FontName','Times New Roman')

legend('R','G','B','Al','Cu','Na')
title('Spectral Response Curve, All
Emission','FontSize',20,'FontName','Times New Roman')

```

## Appendix C: MatCine – a multiplatform MATLAB program to extract raw data from Phantom Camera “.cine” video formats

### *C.1. Introduction*

A functionalized MATLAB program has been developed for the import of the “.cine” video format that the Vision Research Phantom Camera family uses for saving their videos. An old software<sup>296</sup> made by Adam Light in 2013 was originally adopted when starting this, but was largely rewritten for updated file format changes in the Vision Research .cine video format in the early days of writing this dissertation (quite literally starting from day 1). A pyCine\_V2 version was rewritten in Python 3 as a result to update the pyCine with the latest .cine file format, but since videos were largely being processed in MATLAB, it was reformatted into MatCine. In the future, if users or developers need to update this, the new MatCine layout is much easier to edit than the pyCine version and makes data handling much better for later processing purposes. Detailed .cine file format developer documents can be accessed on the Vision Research Developer Zone by request.

## ***C.2. Running MatCine***

The program required input is a .cine video file. The program then executes and asks the user if they would like to crop the video window. If the user chooses to crop the window, then the brightest frame in the video is displayed and the user can create a bounding box to crop the video, right click, and click crop. The program then asks users to confirm the crop window size and proceeds to crop the video. The output of the program is the header information, bitmap, setup information (contains recording settings), and the raw image array.

The last edit of this software was on March 5, 2020. If sections of this code are used in other programs or should be cited as part of future research, please reference this dissertation. An online version has also been stored with version control and is available by reasonable request.

## ***C.3. MATLAB code***

```
function [header, bitmap, setup, raw_image_array]=MatCine(filename)

%% MATCINE - Cine Extraction Software for Matlab
% Dylan J. Kline
% University of Maryland College Park / NASA Goddard Space Flight
Center
% dkline@umd.edu / dylan.kline@nasa.gov

% This program has been designed to extract raw data from a Cine
recorded
% file produced by the Vision Research Phantom Camera Family. It is
based on
% the file format as described in the file 'Cine File Format June
2011' by
% Vision Research.

% Adapted from PyCine_V2 by Adam D. Light and Dylan J. Kline

%% File Selection

% % Pick Cine File
% [files, path]=uigetfile({'*.cine'}, 'Pick a Cine File');
% filename=fullfile(path, char(files));

% Open Cine file
```



```

cine = fopen(filename);

%% Header

header_length = 44;           % first 44 bytes are header

header.Type = fscanf(cine,'%c',2);
header.Headersize = fread(cine,1,'ushort');
header.Compression = fread(cine,1,'ushort');
header.Version = fread(cine,1,'ushort');
header.FirstMovieImage = fread(cine,1,'long');
header.TotalImageCount = fread(cine,1,'uint');
header.FirstImageNo = fread(cine,1,'long');
header.ImageCount = fread(cine,1,'uint');
header.OffImageHeader = fread(cine,1,'uint');
header.OffSetup = fread(cine,1,'uint');
header.OffImageOffsets = fread(cine,1,'uint');
header.TriggerTime = [fread(cine,1,'uint'), fread(cine,1,'uint')];

if ftell(cine)~=header_length
    disp('There has been an error reading the header!');
    return;
end

%% Bitmap Info
bitmapinfo_length = 40;       % next 40 bytes are bitmap header

bitmap.biSize = fread(cine,1,'uint');
bitmap.biWidth = fread(cine,1,'long');
bitmap.biHeight = fread(cine,1,'long');
bitmap.biPlanes = fread(cine,1,'ushort');
bitmap.biBitCount = fread(cine,1,'ushort');
bitmap.biCompression = fread(cine,1,'uint');
bitmap.biSizeImage = fread(cine,1,'uint');
bitmap.biXPelsPerMeter = fread(cine,1,'long');
bitmap.biYPelsPerMeter = fread(cine,1,'long');
bitmap.biClrUsed = fread(cine,1,'uint');
bitmap.biClrImportant = fread(cine,1,'uint');

if ftell(cine)~=header_length+bitmapinfo_length
    disp('There has been an error reading the bitmap info!');
    return;
end

if bitmap.biBitCount~=16
    disp('Your video is not 16bpp unpacked!');
    return;
end

%% Deprecated Skip
deprecated_skip=140;

fseek(cine,deprecated_skip,0);

if ftell(cine)~=header_length+bitmapinfo_length+deprecated_skip
    disp('There has been an error reading the header!');
    return;
end

%% Setup
% setupcheck = fscanf(cine,'%c',2); %%%%%%

```

```

% setup.Length = fread(cine,1,'ushort');
fseek(cine,header.OffSetup,'bof');

setup.Framerate16 = fread(cine,1,'uint16');
setup.Shutter16 = fread(cine,1,'uint16');
setup.PostTrigger16 = fread(cine,1,'uint16');
setup.FrameDelay16 = fread(cine,1,'uint16');
setup.AspectRatio = fread(cine,1,'uint16');
setup.Res7 = fread(cine,1,'uint16');
setup.Res8 = fread(cine,1,'uint16');
setup.Res9 = fread(cine,1,'uint8');
setup.Res10 = fread(cine,1,'uint8');
setup.Res11 = fread(cine,1,'uint8');
setup.TrigFrame = fread(cine,1,'uint8');
setup.Res12 = fread(cine,1,'uint8');
setup.DescriptionOld = fscanf(cine,'%c',121); %%%%%%%%%
setup.Mark = fscanf(cine,'%c',2); %%%%%%%%%;
setup.Length = fread(cine,1,'uint16');
setup.Res13 = fread(cine,1,'uint16');
setup.SigOption = fread(cine,1,'uint16');
setup.BinChannels = fread(cine,1,'int16');
setup.SamplesPerImage = fread(cine,1,'uint8');
setup.BinName = fscanf(cine,'%c',88); %%%%%%%%%
setup.AnalOptions = fread(cine,1,'uint16');
setup.AnalChannels = fread(cine,1,'int16');
setup.Res6 = fread(cine,1,'uint8');
setup.AnalBoard = fread(cine,1,'uint8');
setup.ChOption = fread(cine,8,'int16');
setup.AnalGain = fread(cine,8,'float');
setup.AnalUnit = fscanf(cine,'%c',48); %%%%%%%%%
setup.AnalName = fscanf(cine,'%c',88); %%%%%%%%%
setup.lFirstImage = fread(cine,1,'int32');
setup.dwImageCount = fread(cine,1,'uint32');
setup.nQFactor = fread(cine,1,'int16');
setup.wCineFileType = fread(cine,1,'uint16');
setup.szCinePath = fscanf(cine,'%c',260); %%%%%%%%%
setup.Res14 = fread(cine,1,'uint16');
setup.Res15 = fread(cine,1,'uint8');
setup.Res16 = fread(cine,1,'uint8');
setup.Res17 = fread(cine,1,'uint16');
setup.Res18 = fread(cine,1,'double');
setup.Res19 = fread(cine,1,'double');
setup.Res20 = fread(cine,1,'uint16');
setup.Res1 = fread(cine,1,'int32');
setup.Res2 = fread(cine,1,'int32');
setup.Res3 = fread(cine,1,'int32');
setup.ImWidth = fread(cine,1,'uint16');
setup.ImHeight = fread(cine,1,'uint16');
setup.EDRShutter16 = fread(cine,1,'uint16');
setup.Serial = fread(cine,1,'uint32');
setup.Saturation = fread(cine,1,'int32');
setup.Res5 = fread(cine,1,'uint8');
setup.AutoExposure = fread(cine,1,'uint32');
setup.bFlipH = fread(cine,1,'uint32');
setup.bFlipV = fread(cine,1,'uint32');
setup.Grid = fread(cine,1,'uint32');
setup.FrameRate = fread(cine,1,'uint32');
setup.Shutter = fread(cine,1,'uint32');
setup.EDRShutter = fread(cine,1,'uint32');
setup.PostTrigger = fread(cine,1,'uint32');
setup.FrameDelay = fread(cine,1,'uint32');
setup.bEnableColor = fread(cine,1,'uint32');
setup.CameraVersion = fread(cine,1,'uint32');
setup.FirmwareVersion = fread(cine,1,'uint32');

```

```

setup.SoftwareVersion = fread(cine,1,'uint32');
setup.RecordingTimeZone = fread(cine,1,'int32');
setup.CFA = fread(cine,1,'uint32');
setup.Bright = fread(cine,1,'int32');
setup.Contrast = fread(cine,1,'int32');
setup.Gamma = fread(cine,1,'int32');
setup.Res21 = fread(cine,1,'uint32');
setup.AutoExpLevel = fread(cine,1,'uint32');
setup.AutoExpSpeed = fread(cine,1,'uint32');

setup.AutoExpRect.left = fread(cine,1,'int32'); %%%
setup.AutoExpRect.top = fread(cine,1,'int32'); %%%
setup.AutoExpRect.right = fread(cine,1,'int32'); %%%
setup.AutoExpRect.bottom = fread(cine,1,'int32'); %%%

setup.WBGain1.R = fread(cine,1,'float'); %%%
setup.WBGain1.B = fread(cine,1,'float'); %%%
setup.WBGain2.R = fread(cine,1,'float'); %%%
setup.WBGain2.B = fread(cine,1,'float'); %%%
setup.WBGain3.R = fread(cine,1,'float'); %%%
setup.WBGain3.B = fread(cine,1,'float'); %%%
setup.WBGain4.R = fread(cine,1,'float'); %%%
setup.WBGain4.B = fread(cine,1,'float'); %%%

setup.Rotate = fread(cine,1,'int32');

setup.WBView.R = fread(cine,1,'float'); %%%
setup.WBView.B = fread(cine,1,'float'); %%%

setup.RealBPP = fread(cine,1,'uint32');
setup.Conv8Min = fread(cine,1,'uint32');
setup.Conv8Max = fread(cine,1,'uint32');
setup.FilterCode = fread(cine,1,'int32');
setup.FilterParam = fread(cine,1,'int32');

setup.UF.dim = fread(cine,1,'int32'); %%%
setup.UF.shifts = fread(cine,1,'int32'); %%%
setup.UF.bias = fread(cine,1,'int32'); %%%
setup.UF.Coef = fread(cine,25,'int32'); %%%

setup.BlackCalsVer = fread(cine,1,'uint32');
setup.WhiteCalsVer = fread(cine,1,'uint32');
setup.GrayCalsVer = fread(cine,1,'uint32');
setup.bStampTime = fread(cine,1,'uint32');
setup.SoundDest = fread(cine,1,'uint32');
setup.FRPSteps = fread(cine,1,'uint32');
setup.FRPImgNr = fread(cine,16,'int32');
setup.FRPRate = fread(cine,16,'uint32');
setup.FRPExp = fread(cine,16,'uint32');
setup.MCCnt = fread(cine,1,'int32');
setup.MCPercent = fread(cine,64,'float');
setup.CICalib = fread(cine,1,'uint32');
setup.CalibWidth = fread(cine,1,'uint32');
setup.CalibHeight = fread(cine,1,'uint32');
setup.CalibRate = fread(cine,1,'uint32');
setup.CalibExp = fread(cine,1,'uint32');
setup.EDR = fread(cine,1,'uint32');
setup.CalibTemp = fread(cine,1,'uint32');
setup.HeadSerial = fread(cine,4,'uint32');
setup.RangeCode = fread(cine,1,'uint32');
setup.RangeSize = fread(cine,1,'uint32');
setup.Decimation = fread(cine,1,'uint32');

```

```

setup.MasterSerial = fread(cine,1,'uint32');
setup.Sensor = fread(cine,1,'uint32');
setup.ShutterNs = fread(cine,1,'uint32');
setup.EDRShutterNs = fread(cine,1,'uint32');
setup.FrameDelayNs = fread(cine,1,'uint32');
setup.ImPosXAcq = fread(cine,1,'uint32');
setup.ImPosYAcq = fread(cine,1,'uint32');
setup.ImWidthAcq = fread(cine,1,'uint32');
setup.ImHeightAcq = fread(cine,1,'uint32');
setup.Description = fscanf(cine,'%c',4096); %%%%%%%%%
setup.RisingEdge = fread(cine,1,'uint32');
setup.FilterTime = fread(cine,1,'uint32');
setup.LongReady = fread(cine,1,'uint32');
setup.ShutterOff = fread(cine,1,'uint32');
setup.Res4 = fread(cine,16,'uint8');
setup.bMetaWB = fread(cine,1,'uint32');
setup.Hue = fread(cine,1,'int32');
setup.BlackLevel = fread(cine,1,'int32');
setup.WhiteLevel = fread(cine,1,'int32');
setup.LensDescription = fscanf(cine,'%c',256); %%%%%%%%%
setup.LensAperture = fread(cine,1,'float');
setup.LensFocusDistance = fread(cine,1,'float');
setup.LensFocalLength = fread(cine,1,'float');
setup.fOffset = fread(cine,1,'float');
setup.fGain = fread(cine,1,'float');
setup.fSaturation = fread(cine,1,'float');
setup.fHue = fread(cine,1,'float');
setup.fGamma = fread(cine,1,'float');
setup.fGammaR = fread(cine,1,'float');
setup.fGammaB = fread(cine,1,'float');
setup.fFlare = fread(cine,1,'float');
setup.PedestalR = fread(cine,1,'float');
setup.PedestalG = fread(cine,1,'float');
setup.PedestalB = fread(cine,1,'float');
setup.fChroma = fread(cine,1,'float');
setup.ToneLabel = fscanf(cine,'%c',256); %%%%%%%%%
setup.TonePoints = fread(cine,1,'int32');
setup.fTone = fread(cine,64,'float'); %%%%%%%%%
setup.UserMatrixLabel = fscanf(cine,'%c',256); %%%%%%%%%
setup.EnableMatrices = fread(cine,1,'uint32');
setup.fUserMatrix = fread(cine,9,'float'); %%%%%%%%%
setup.EnableCrop = fread(cine,1,'uint32');

setup.CropRect.left = fread(cine,1,'int32'); %%%%
setup.CropRect.top = fread(cine,1,'int32'); %%%%
setup.CropRect.right = fread(cine,1,'int32'); %%%%
setup.CropRect.bottom = fread(cine,1,'int32'); %%%%

setup.EnableResample = fread(cine,1,'uint32');
setup.ResampleWidth = fread(cine,1,'uint32');
setup.ResampleHeight = fread(cine,1,'uint32');
setup.fGain16_8 = fread(cine,1,'float');
setup.FRPShape = fread(cine,16,'uint32');

% couldnt get time to cooperate
fseek(cine,8,0)

setup.fPbRate = fread(cine,1,'float');
setup.fTcRate = fread(cine,1,'float');
setup.CineName = fscanf(cine,'%c',256); %%%%%%%%%

%% Read Images
fseek(cine,header.OffImageOffsets,'bof');

```

```

raw_image_array=zeros([bitmap.biHeight,bitmap.biWidth,header.ImageCou
nt],'uint16');
pointer_array=fread(cine,header.ImageCount,'uint64');
fseek(cine,pointer_array(1),'bof');
annotation_size=fread(cine,1,'uint');
string_size=annotation_size-8;
annotation=fscanf(cine,'%c',string_size);
image_size=fread(cine,1,'uint');

if image_size~=2*bitmap.biWidth*bitmap.biHeight
    disp('Image sizes are not correct, file may be corrupt or using
an outdated version of PCC!')
    return;
end

wait=waitbar(0,sprintf('Reading %d images...',header.ImageCount));
for i=1:header.ImageCount

    raw_image_array(:, :, i)=rot90(reshape(fread(cine,image_size/2,'ushort'
),[bitmap.biWidth,bitmap.biHeight]));
    fseek(cine,annotation_size,0);
    waitbar(i/header.ImageCount)
end
close(wait)

%% Cropping Video
% This option is added is as a way for the user to consolidate
memory on
% their computer by cropping the window width.

crop = input('Would you like to crop the video (y/n)? ','s');

% Bayer Pattern - need full gbrg square for correct cropping. must
be even
% height and width, starting on an odd number x and y.
%   1 2 3 4 5 6 ...
% 1| g b g b g b ...
% 2| r g r g r g ...
% 3| g b g b g b ...
% 4| r g r g r g ...
% 5| g b g b g b ...
% 6| r g r g r g ...

if crop=='y'

    % making a normalized light intensity calculation
    for i=1:header.ImageCount
        % Integrated amount of light hitting the sensor
        light_intensity(i)=sum(sum(sum(raw_image_array(:, :, i))));
    end
    light_intensity=light_intensity/max(light_intensity);
    [~,brightframe]=max(light_intensity);

    gain=4096/max(max(raw_image_array(:, :, brightframe)));

    [~,rect2]=imcrop(image(double(demosaic(raw_image_array(:, :, brightfram
e)*gain,'gbrg'))/4096));
    close all

    xmin=2*round((rect2(1)+1)/2)-1;
    newwidth=2*round(rect2(3)/2);

```

```

ymin=2*round((rect2(2)+1)/2)-1;
newheight=2*round(rect2(4)/2);

crop=input(sprintf('Area selected is %d x %d, would you like to
change it (y/n)? ',newheight,newwidth),'s');
if crop=='y'
    newheight=2*round(input('Desired height: ')/2);
    newwidth=2*round(input('Desired width: ')/2);
end

rect2=[xmin ymin newwidth newheight];

raw_image_array=raw_image_array(rect2(2):(rect2(2)+rect2(4)-
1),rect2(1):(rect2(1)+rect2(3)-1),:);

else
    return;
end

% %% Save Data
% save(char(sprintf('%s.mat',filename(1:end-
4))),'raw_image_array','header');

fclose('all')

end

```

## Appendix D: pyroCam – A MATLAB script for color camera

### pyrometry

This appendix will serve as the user manual for the high-speed color camera pyrometry code as has been employed in the research originally described in Chapter 2. The code was designed as a use for high-speed videography experiments where temperature of the reaction can provide useful insight on the mechanisms or heat transfer rates. This code can be run using any of the color cameras that the Zachariah Group has as of December 2020. Throughout the manual, text in `this font` will represent variables or expressions that are used in a MATLAB command.

The manual below is broken up into different sections on (1) running the pyrometry code, (2) input and output information for the program, (3) detailed variable listing and descriptions, (4) generalized code structure, (5) a summary, (6) version history notes if applicable, and (7) the raw MATLAB code.

#### ***D.1. Running pyroCam***

The executable MATLAB script can be found in the Zachariah Group Drive at Volumes/users/Kline/Codes/pyroCam/kline\_temperature\_colorcam10.m. An online version has also been stored with version control and is available by reasonable request. The code has numerous dependencies on installed packages for MATLAB and those that have been written by the author and stored in Volumes/users/Kline/Codes/pyroCam/.

***Table D-1:*** Dependencies for temperature measurement code.

Name	Description
------	-------------

MATLAB Image Processing Toolbox	General use toolbox which contains required functions for image processing related purposes. For this project in particular, regionprops is required.
MatCine	Used to extract raw image array from .cine file formats.

The opening lines of `kline_temperature_colorcam10.m` attempts to automatically add the required external functions but if this fails it can always be done manually by right-clicking on the External Functions folder and choosing Add to Path>Selected Folder and Subfolders.

To run `kline_temperature_colorcam10.m`, open the folder in the script viewer in MATLAB and click Run in the Editor Ribbon. The program will execute automatically from the first written line. Alternatively, users can choose to Run Section in the script viewing window since the code has been split up by section.

## ***D.2. Description of input and output***

The following parameters are used as input and outputs for the pyrometry code and the prompts with their expected values are described below. Users will typically have to experimentally determine values for the different options if they find that the initial results are not satisfactory (i.e. poor choice of color bar limits, unusual temperature measurements).

### ***Input***

**Table D-2:** Input options and prompts for pyrometry code with associated descriptions.

<b>Prompt</b>	<b>Description</b>
Pick Cine File (GUI)	Users are asked to navigate to the desired .cine file that they would like to be analyzed using the graphical user interface of their machine. The program will automatically save the path



	<p>that the user has navigated to for future use in saving and organizing files.</p> <p>Users may also choose to pick a .mat output file and the program will execute normally, though this feature is rarely used and it would be better to simply pick the .cine file.</p> <p>Variables stored:  files, path, filename, pathstr, name, ext, header, bitmap, setup, raw_image_array, im_height, im_width, num_pixels, num_images</p>
Framerate (pps) :	<p>If a user picked a .mat file without the time information, users will need to input the framerate. When loading a .cine file, this prompt is not shown since the framerate is automatically extracted from the header and setup files produced by MatCine.</p> <p>Input must be a number.</p> <p>Variables stored:  framerate</p>
Was the video rerecorded and decimated?:	<p>If a user has found that a video of theirs was too large to save initially and then went into the Phantom Camera Control software and decimated the video (i.e. saving 1 of every 10 frames), unfortunately the timestamps would no longer be correct. This option fixes that problem in the code.</p> <p>Input options are y or n.  If y is selected, the user will be asked by what factor it was decimated. Timestamps are then calculated using the decimating factor.  If n is selected, the time is calculated normally using the framerate.  If something other than y or n is selected, then the program will execute as if n was chosen.</p> <p>Variables stored:  decimated, time</p>
Decimated by:	<p>If a user chose y in the above prompt, then they will be asked by which factor their video was decimated. Timestamps will then be calculated using the video framerate and the decimation factor using the equation below where t is time, n</p>

	<p>is the image number, f is the framerate (in frames/s) and d is the decimation factor. Default value is 1.</p> $t = \frac{n}{f * d}$ <p>Input must be an integer.</p> <p>Variables stored: decimated, time</p>
Lambda Correction (y/n) :	<p>User option for calibration curve to be used for temperature measurements. The theory of the lambda correction is described in Chapter 2. A major assumption in color camera pyrometry is the graybody assumption which states that the emissivity of the samples is independent of wavelength. However, it has been found that a better approximation for aluminum containing samples with a thin flame front is that emissivity is related to the inverse of wavelength (<math>\epsilon \sim \lambda^{-1}</math>). As such, there is a calibration option where the ratios that will be matched will have employed the “lambda correction” in the calculated predicted ratios.</p> <p>The lambda correction should be primarily used for samples that have a thin flame front and contain aluminum. If users are unsure which to choose, it would be better to not use the lambda correction. For samples that have an optically thick image they are observing (i.e. shining a light through the sample will not impact the image), the lambda correction should not be employed. For videos taken with the microscope at high magnifications, this should also not be employed.</p> <p>As a general observation, it has been seen that using the lambda correction will generate temperatures ~200-300 K lower than those if it was not employed.</p> <p>Input options are y, n, t, or h. If y is selected, the calibration ratios for matching are changed in the line following the input (ratios_MiroM110_773-4773_lambdacorrect.mat). If n is selected, the original calibration file is used (ratios_MiroM110_773-4773.mat). If t is selected, a lambda corrected calibration curve with a higher upper temperature limit (5773 K) is selected</p>

	<p>(ratios_MiroM110_773-5773_lambdacorrect.mat). This will also widen the restrictions on the ratio thresholding bounds.</p> <p>If h is selected, the calibration ratios for a non-lambda corrected value are selected and the ratio thresholding bounds used for displaying images in the code are widened to a point that they are essentially eliminated.</p> <p>If y, n, t, or h are selected, the program will execute as if n was chosen.</p> <p>Older versions of the code may have an option for m which was the lambda corrected calibration curve for the high-speed microscopy apparatus including the Nikon 40x objective lens, but this option has been removed since the apparatus is no longer in use.</p> <p>Variables stored: lambdacorrect, ratios_summary</p>
<p>Choose lens for calibration factors Micro (1), Macro (2), or Wide Angle (3):</p>	<p>User option to select the lens that was used for the experiments. The microscope lens is an InfinityOptics K2 DistaMax lens equipped with an NTX tube and a CF4 objective. The macro lens is a Nikon AF Micro Nikkor 105mm 1:2.8 D lens. The wide-angle lens is a Nikon AF Nikkor 28mm 1:2.8 D lens.</p> <p>Input options are 1, 2, or 3.</p> <p>If 1 is selected, the calibration factors for the Infinity Optics K2 DistaMax lens is chosen.</p> <p>If 2 is selected, the calibration factors for the Nikon macro lens are selected.</p> <p>If 3 is selected, the calibration factors for the Nikon wide angle lens are selected.</p> <p>If a number other than 1, 2, or 3 is selected, the program will generate a message saying that the lens option was not recognized and will halt execution.</p> <p>Variables stored: calibFactors</p>
<p>Error Threshold (0.1 default):</p>	<p>User option to change the error thresholding that will be used for mean/median temperature measurements and for intentional covering in generated filtered videos. The error threshold is calculated by summing the error in the three calculated channel ratios to their values at the determined temperature in the calibration curve. If the summed error is</p>

	<p>greater than the error threshold, the temperature will be excluded from being shown in the videos and will instead appear as a copper color.</p> <p>A good first basis should be 0.1 which would nominally generate an average frame error of ~110 K. A check was also performed on an error threshold value of 0.2 which nominally generated a ~200 K error, but some errors were as high as 400 K when the video was very dim. If users wanted to effectively eliminate the ratio thresholding, they would choose a very high number for the error threshold.</p> <p>Input options must be an integer. If a user accidentally fails to enter a number, a default number of 0.1 will be chosen.</p> <p>Variables stored:  <code>filters.errorThresh</code></p>
<p>Select bright reaction front? (y/n) :</p>	<p>User option select a bright reaction front in the image and only acquire temperature for regions that were in that bright reaction front. Oftentimes there would be films that would burn on a glass slide and, in the background of the image, the plume would read as valid temperature measurements when in reality the temperature was far from the object being focused on. There are some other scenarios that this section might be useful, too. Essentially, this gives the user the option to binarize and modify their temperature maps based on brightness.</p> <p>Input options are y or n.          If y is selected, the user will be prompted regarding binary thresholding options.          If n is selected, code will not mask “dark” regions.          If something other than y or n is selected, the program will execute as if n was chosen.</p> <p>Variables stored:  <code>filters.front</code></p>
<p>Binary threshold (0.25-0.4 recommended) :</p>	<p>If the user chooses to select the bright reaction front in the video, they will be prompted with the option to binarize their videos with a certain binarization factor. Values have been suggested empirically, though this section will likely need to be run a few times for users wishing to employ this option.</p> <p>Input must be an integer.</p>

	<p>Variables stored:  <code>bw_thresh, filters.front</code></p>
<p>Choose colormap limits or define your own  Auto (1) or Custom (2):</p>	<p>User option to choose their own colormaps using an auto colormap option or a custom colormap option. As a first option, users might choose to use the auto option which gives a range that is 0.85-1.15x of the median value of the median frame temperatures in the video. Otherwise, it is recommended that users who wish to publish the generated images choose the custom option to make the limits round numbers.</p> <p>Input options are 1 or 2.  If 1 is selected, the auto colormap will be employed as described above.  If 2 is selected, user will be prompted to enter their colormap limits as an array.  If something other than 1 or 2 is selected, the program will execute as if 2 was chosen.</p> <p>Files generated:  <code>vidT.cmapchoice, vidT.cmaplim</code></p>
<p>Enter colormap limits as array:</p>	<p>If the user chose to enter their own colormap limits, they must enter them as an array.</p> <p>Input must be an array in the form of  <code>[lower_limit upper_limit]</code>  where the <code>lower_limit</code> and <code>upper_limit</code> values are integers.</p> <p>Variables stored:  <code>vidT.cmaplim</code></p>
<p>Define video length based on time (1) or framerate (2)?:</p>	<p>User video options to determine the framerate of the generated temperature videos based on the length of the video to have or the framerate they would like the video to have.</p> <p>Input options are 1 or 2.  If 1 is selected, the user will be prompted to input the desired length of their video.  If 2 is selected, user will be prompted to enter their desired video framerate.  If something other than 1 or 2 is selected, the program will execute as if 2 was chosen.</p>

	<p>Variables stored: vid.framerateortime</p>
Desired length of video (s) :	<p>If the user chose to define the video timing based on the desired length of the video, they will be prompted to enter their desired length of the video in seconds.</p> <p>Input must be an integer.</p> <p>Variables stored: vid.vidtime</p>
Desired framerate (frames/s) :	<p>If the user chose to define the video timing based on the framerate of the final video, they will be prompted to enter their framerate of the video in frames/second. For general purposes, a value between 30-60 frames/s is acceptable.</p> <p>Input must be an integer.</p> <p>Variables stored: vid.enteredframerate, vid.vidtime</p>
Steps between frames (1 for no skipped frames) :	<p>Regardless if the user chose to define their video based on the total video length or framerate, they will be asked if they would like to decimate their videos. If a user would like to display every image in their video, they would enter 1, however if they would like to show every 10 images (for long videos or first run), they would enter 10.</p> <p>Input must be an integer.</p> <p>Variables stored: vid.imstep</p>
Show gray (out of bounds) pixels? (y/n) :	<p>This option will mask videos with a gray pixel where the ratios that were calculated for the pixel were outside the ratio thresholding bounds. This was implemented because some videos would show dark spots in videos that were difficult to explain, but this was a rare occurrence. A notable reason that the ratios would be far out of bounds would be from color skewing by species that emit light at very specific frequencies (i.e. sodium) or absorb light when concentrated (i.e. iodine).</p> <p>Input options are y or n.</p>

	<p>If y is selected, a gray mask will be applied for pixels that are outside the bounds of the ratio thresholding.</p> <p>If n is selected, the pixels outside of the error thresholding will appear black.</p> <p>If something other than y or n is selected, the program will execute as if n was chosen.</p> <p>Variables stored: bg.rgbRatio</p>
Is the image rotated? (y/n) :	<p>If users recorded videos with the camera rotated, the videos would also appear rotated. This option would allow those users to correct their videos for the rotation.</p> <p>Input options are y or n.</p> <p>If y is selected, users will be prompted to provide more information on the rotation.</p> <p>If n is selected, the video will not be rotated.</p> <p>If something other than y or n is selected, the program will execute as if n was chosen.</p> <p>Variables stored: vid.rotk</p>
Number of 90-degree clockwise rotations to apply to image:	<p>If users chose to rotate their videos in the previous prompt, they are asked to provide the number of rotations they applied to the camera when recording (or would like to be applied to the video).</p> <p>Input must be an integer.</p> <p>Variables stored: vid.rotk</p>
Video gain auto (1) or custom (2)?:	<p>User option to fix aesthetics of a video for filtered temperature videos. If a video is dark, users may wish to apply a higher than normal gain. This will not impact the temperature measurements.</p> <p>Input options are 1 or 2.</p> <p>If 1 is selected, the gain will be automatically calculated such that the brightest pixel in the brightest frame is made to be white.</p> <p>If 2 is selected, user will be shown the calculated autogain value and will be prompted to enter a new gain.</p>

	<p>If something other than 1 or 2 is selected, the program will execute as if 2 was chosen.</p> <p>Variables stored: vid.gain</p>
Desired gain:	<p>If a user chose to use a custom gain for their video, they will be shown the calculated autogain value and will be asked to enter a new video. As a suggestion, typically 1.5-2x the autogain value would be good for presentations and publishing.</p> <p>Input must be a number.</p> <p>Variables stored: vid.gain</p>
Would you like a temperature only video? (y/n) :	<p>User option to generate an mp4 video containing only the false color temperature images. Images generated for the video will be filtered where white denotes pixels where oversaturation occurred, copper colored pixels have summed errors larger than the error threshold, and gray colored pixels are outside of the ratio thresholding bounds.</p> <p>Input options are y or n. If y is selected, a video will be generated. If n is selected, the section of code is skipped. If something other than y or n is selected, the program will execute as if n was chosen.</p> <p>Variables stored: ogvid</p> <p>Files generated: &lt;filename&gt;.mp4</p>
Would you like a side-by-side T video? (y/n) :	<p>User option to generate an mp4 video containing the false color temperature images alongside the original color images. Images generated for the video will be filtered where white denotes pixels where oversaturation occurred, copper colored pixels have summed errors larger than the error threshold, and gray colored pixels are outside of the ratio thresholding bounds. This is the most common video that is generated since it shows both the original videos and the temperature videos.</p>



	<p>Input options are y or n.  If y is selected, a video will be generated.  If n is selected, the section of code is skipped.  If something other than y or n is selected, the program will execute as if n was chosen.</p> <p>Variables stored:  filtervid</p> <p>Files generated:  &lt;filename&gt;_filtered.mp4</p>
Would you like to clean up directories? (y/n) :	<p>User option to create a new folder with all of the generated files in the newly created folder. Folder will have name format of yyyyMMdd &lt;filename&gt; in the directory that the original video file was stored. The generated .mat files, videos, figure files, and original video file will be moved into the folder.</p> <p>Input options are y or n.  If y is selected, a folder will be made and files generated during the code will be moved into that folder.  If n is selected, the code section is skipped.  If something other than y or n is selected, the program will execute as if n was chosen.</p>

### Output

**Table D-3:** Output files for pyrometry code with associated descriptions.

<b>Output</b>	<b>Description</b>
<filename>.mat	MATLAB output file containing all of the variables generated when running the code
<filename>_maxmean.fig	Figure file showing the mean and median temperatures of each frame. Mean and median values are calculated using regions of at least 10 contiguous pixels that are not saturated, within the error threshold, and within the ratio threshold.
<filename>.mp4	Optional video file containing only temperature images.

<code>&lt;filename&gt;_filtered.fig</code>	Optional video file containing a side-by-side of the original video and temperature image.
--	--

### ***D.3. Variable description***

A full list of variables, their data types, and their descriptions is provided below for reference when making modifications to the code or debugging the program. The variables are listed in alphabetical order.

**Table D-4:** Variable list and description for pyrometry code.

<b>Variable</b>	<b>Description</b>
<code>aspect</code>	Aspect ratio of generated figure.
<code>ax</code>	Axis handle.
<code>ax1</code>	Axis handle.
<code>ax2</code>	Axis handle.
<code>AxesHandle</code>	Axis handle.
<code>B</code>	Blue pixel values.
<code>basename</code>	Name of video selected without extension.
<code>bg.err</code>	Copper background used for temperature map making.
<code>bg.errIm</code>	Error colored pixel image handle for video creation.
<code>bg.ratio</code>	Gray background used for temperature map making.
<code>bg.sat</code>	White background used for temperature map making.
<code>bg.satIm</code>	Saturated pixel image handle for video creation.
<code>bg.zero</code>	Black background used for temperature map making.
<code>bg.zeroIm</code>	Dark pixel image handle for video creation.
<code>bitmap</code>	Bitmap output from MatCine
<code>brightframe</code>	Index of the brightest frame in the video.
<code>bwthresh</code>	User-entered thresholding to be used if the user chooses to selectively report data for the bright reaction front.
<code>calibchoice</code>	User defined calibration choice between microscope, macro lens, and wide angle lens.
<code>calibFactors</code>	Calibration factor array for user-chosen lens. [Cgr, Cbg, Cbr].
<code>cleanup</code>	User input option to cleanup directories.

<code>decimated</code>	User-entered value describing by which factor their video was rerecorded and decimated.
<code>demosaicIm</code>	Image returned after being demosaiced with the built in MATLAB routine. Size is <code>im_height</code> x <code>im_width</code> x 3 for red ( <code>(:,:,1)</code> ), green ( <code>(:,:,2)</code> ), and blue ( <code>(:,:,3)</code> ) pixels.
<code>dmMat</code>	Reshaped image array of with format of [pixel#, G/R calibration corrected ratio, B/G calibration corrected ratio, and B/R calibration corrected ratio] at each pixel for a given frame.
<code>ext</code>	Extension of file chosen in graphical user interface.
<code>f</code>	Figure handle.
<code>fig</code>	Figure handle.
<code>fig_height</code>	Figure height.
<code>fig_width</code>	Figure width.
<code>figdim</code>	Calculated figure dimension handle.
<code>filename</code>	Selected video filename.
<code>files</code>	Specifies file selected by user for processing from graphical user interface.
<code>filters.blacklevel</code>	Default black level of camera (64).
<code>filters.blacklevelfactor</code>	Factor for increasing the black level in the video for the <code>im_construct</code> variable.
<code>filters.contiguous</code>	Number of contiguous pixels required to be counted as part of mean/median temperature measurements.
<code>filters.error</code>	Boolean variable which denotes whether or not the error in the corresponding pixel was above the <code>TErrorThresh</code> .
<code>filters.errorThresh</code>	User entered value for thresholding to determine whether or not a pixel was high in error. Generally 0.1-0.2 should be used.
<code>filters.front</code>	Boolean variable which denotes whether or not a pixel is part of the bright reaction front. If the user did not choose to selectively show data for the bright reaction front, then this is just an array of 1s.
<code>filters.lightThresh</code>	Lower bound of the normalized light intensity to be valid for calculation of mean/median temperature.
<code>filters.nan</code>	Boolean variable of size image height x image width x frame which determines

	whether or not the calculated temperature was a NaN.
<code>filters.ratio</code>	Boolean variable which denotes whether or not the calculated color ratios is within the acceptable ratio thresholds.
<code>filters.ratioThresh</code>	Array describing the lower and upper limits that are acceptable for ratios in order to be matched. Originally added for data processing improvements, but should be widened if there are significant areas of gray pixels.
<code>filters.satlevel</code>	Value to be used as thresholding for saturation in the videos. Set to 4000.
<code>filters.summary</code>	Simple variable that is temporarily stored to show which pixels are above the zero threshold, below the saturation threshold, within the error thresholding, and considered to be a relevant part of the picture (if applicable). Used for calculation of mean and median temperatures.
<code>filters.T_low</code>	Lowest temperature threshold.
<code>filters.zero</code>	Variable which shows where the <code>im_construct</code> was 0.
<code>filtervid</code>	User input used to determine whether to generate side-by-side temperature video.
<code>frameno</code>	Frame number label handle used when generating videos.
<code>framerate</code>	Framerate either determined from setup values in MatCine or user entered.
<code>G</code>	Green pixel values.
<code>header</code>	Header output from MatCine
<code>i</code>	Counter variable.
<code>im_height</code>	Image height.
<code>im_width</code>	Image width.
<code>imSize</code>	Variable describing height and width of the video being analyzed.
<code>k</code>	Counter variable.
<code>lambdacorrect</code>	User option to correct emissivity for wavelength dependency, but also has options to extend calibration temperature range or loosen the ratio thresholding.
<code>light_intensity</code>	Calculated light intensity made by summing the values of <code>raw_image_array</code> in each frame. Normalized to have the brightest frame = 1.

name	Name of file selected.
num_images	Number of images in file selected.
num_pixels	Number of pixels in a given frame.
ogvid	User input to determine whether to generate temperature only video.
orient	Video orientation.
origAx1Size	Original axis size of axis handle 1.
origAx2Size	Original axis size of axis handle 2.
path	Path of selected video.
pathstr	String of path for selected video.
R	Red pixel values.
ratios_summary	Calibration curve variable automatically loaded when the user chooses which calibration setting they would like to use.
raw_image_array	Three-dimensional array of pixels, image height x image width x frame.
se	Expansion factor for pixels that are saturated or below the black level. Set to 3 empirically.
setup	Setup variable returned from MatCine.
T	Temporary variable which shows the temperature of a single frame during temperature matching calculations.
t2	Temporary variable used for calculation of mean, median temperatures.
Tdm	Reshaped array of calculated calibration curve values with format of [G/R expected ratio, B/G expected ratio, B/R expected ratio] for each temperature (dimension 3).
temp	Temporary variable used throughout code.
TError	Variable which shows the error in the matching for each frame.
TErrorAve	Average error in the different color ratios for a given frame.
TErrorStack	Stack of summed error at each pixel recorded during matching. Size of im_height x im_width x num_images.
Ti	Temporary value used for calculation which returns the index of the minimum error for the pixels during matching within the calibration curve used.
Tim3	Scaled temperature image handle for video creation.
Timage	Three-dimensional array of temperatures as matched to the calibration curve, image

	height x image width x frame. The values are not filtered.
time	Array of timestamps for the given image.
timelab	Time label handle used when generating videos.
Tmax	Array of maximum temperatures as a function of frame.
Tmean	Array of average temperatures as a function of frame for regions that are part of 10 contiguous pixels within the error thresholding, ratio thresholding, and are unsaturated.
Tmedian	Array of median temperatures as a function of frame for regions that are part of 10 contiguous pixels within the error thresholding, ratio thresholding, and are unsaturated.
v	Video handle.
vid.enteredframerate	User-entered framerate to be used when generating temperature videos.
vid.framerateortime	User option to define their video length based on a user entered framerate or total video length.
vid.gain	Gain for video to be applied for original color images. Either automatically calculated or user entered.
vid.imstep	Number of steps between images to be used when generating video.
vid.ratioIm	Gray pixel image handle for video creation.
vid.ratioThresh	Variable introduced to expedite temperature measurements by imposing a threshold on the color ratios that would be unrealistic, thus reducing the number of points that need to be matched. If a calculated ratio is outside of the bounds listed here, then it will be obscured from view in the temperature measurement videos generated.  [Cgr expected lower, Cbg expected lower, Cbr expected lower; Cgr expected upper, Cbg expected upper, Cbr expected upper].
vid.rotk	Number of clockwise rotations to be applied to image.
vid.vidtime	Total time of video.
vidT.c	Colormap scale bar.

<code>vidT.cmapchoice</code>	User input value for colormap limit choice to be determined automatically or put in manually.
<code>vidT.cmaplim</code>	User entered colormap limits [lower_limit upper limit].
<code>vidT.myColorMap</code>	Generated colormap.
<code>vidT.rgbError</code>	RGB values for the color to be used in high-error pixels.
<code>vidT.rgbRatio</code>	RGB values for the color to be used in pixels that are outside of the ratio thresholding.
<code>wait</code>	Wait bar handle.

#### ***D.4. Code structure***

The pyrometry code is broken up into discrete sections as outlined in the code itself and separated by `%%`. A description of each of these sections and their main functions is below, but the program has been heavily commented for further reference.

**Table D-5:** Code structure for pyrometry code.

<b>Section</b>	<b>Description</b>
Introduction and organization	Clearing entire workspace and adding folders/subfolders for dependencies to the working path.
1. Import of images using MatCine	Importing images using MatCine and writing variables for later calculations.
2. Calibration curve information	The initial section of the code that gives the users prompts about the type of video they recorded and its contents. This section allows a user to select to use a modified emissivity behavior, extend the temperature calibration range, or widen limits imposed on predicted ratio thresholds. Using a different emissivity correlation will change temperature measurements.
3. Temperature calculation	This section of the code is where all of the calculations are made for the temperature images and the corresponding error values for the temperatures. Images are demosaiced, reshaped, and individual pixels are matched to a calibration curve that was imported in section 2. Beyond this section, no significant calculations are made regarding the temperature map.
4. Maximum, Mean, and Median	Section of code that is run to complement and simplify the information generated in section 3. This section runs a routine that identifies contiguous regions of pixels that are within the

Temperature Measurements	error threshold, are within the ratio threshold, and are unsaturated and then calculates statistical information about the frame including the mean, median, and maximum temperature values. This section of code also has a routine that would allow a user to selectively look at regions that are within the bright reaction front as specified in the input table in the previous section.
5. Data Saving	Routine for saving data generated in the workspace prior to video creation.
6. Video Setup for Colormap Limits & Time	Section of code that prompts users to provide settings for the false colored temperature videos they would like to generate as part of the code. Users are able to choose custom gains for the original videos, choose custom colormap limits, define the length of their videos, and make rotations to the videos in this section. Settings recorded in this section are used in the next two sections when generating videos.
7. Temperature Only Video Creation	A section that generates a video of only the false color temperature images.
8. Side-by-side Original Color Video and Filtered Temperature Video	Most commonly executed section of code for creating videos that are generated for publication. The videos generated in this section have a side-by-side image of the original color image as captured using the high-speed camera and its corresponding temperature map as calculated using the pyrometry code.
7. Clean Up Directories	An option for the user to clean up directories in their working directory by creating a new dated folder and moving all newly generated files into that folder along with the original video that was used for analysis.

### ***D.5. Summary***

This user manual and code description includes the input, output, and structure of the pyrometry code developed for high-speed videography of highly exothermic reacting materials, but the code has been used extensively for high-speed temperature measurements. If sections of this code are used in other programs or should be cited as



part of future research, please reference either this dissertation or R.J. Jacob, D.J. Kline, M.R. Zachariah, High speed 2-dimensional temperature measurements of nanothermite composites, J. Appl. Phys. 123 (2018) 115902. doi:10.1063/1.5021890.

This work has truly been the cornerstone of my thesis and I am so happy to have been able to develop this entire process from start to finish. It has not only made me a better programmer and researcher, but has given me opportunities to branch out and find new types exciting research and methodologies that I did not think I was ever going to encounter. I have also met a lot of great colleagues that have provided input on and used this software for their own research, and for that I am truly grateful.

#### ***D.6. Version history***

A simple version history is presented below for reference and v10 is provided in the following section.

***Table D-6:*** Version history for flame front extraction code.

<b>Version</b>	<b>Date</b>	<b>Description</b>
1	3/2/17	Initial version which relies on PyCine v2 for pixel extraction and performs rudimentary temperature measurements without built in demosaicing algorithm. Video output in AVI without showing original images.
2	3/6/17	Refined v1 for bug fixes.
3	3/22/17	A large improvement over v2 with new code bits provided by Phil Guerieri. A demosaicing algorithm is now used to generate RGB at every pixel and an improved algorithm for ratio matching was also introduced which dramatically reduced computation time. Error thresholding, black level thresholding, ratio thresholding, and saturation thresholding are introduced. A new video format of side-by-side original vs. temperature measurement videos is introduced. The auto cleanup of directories was also introduced in this version.
4	4/18/17	Introduced the feature to measure mean, median, and maximum temperatures as a function of time.
5	4/29/17	Bug fixes.

6	3/22/19	Introduced option to correct for lambda dependencies in emissivity.
6.1	6/14/17	Bug fixes.
6.2	6/30/17	Bug fixes.
7	7/12/17	Bug fixes.
8	7/12/17	Bug fixes.
9	11/15/18	Rearranged sections of code. Added feature for custom colormap definitions and video time definitions. Changed codec to MP4 for temperature only videos. New calibration options for high temperature videos. Gave users options to enter custom error thresholding.
10	8/20/19	Add a readout about disclaimer and uses. Added an automatic option for colormap limits. Reorganized side-by-side video layout. Removed vertical video option.
	4/27/20	Substantial data reorganization to improve readability of code for future users. Latest version at time of graduation. Code published in private GitHub repository.

### ***D.7. MATLAB code***

```

%% Time-Resolved High Speed Color Camera Temperature Measurements
% Dylan J. Kline
% Chemical Engineering - Zachariah Research Group
% University of Maryland College Park
% dkline@umd.edu

% UNAUTHORIZED DUPLICATION AND/OR REDISTRIBUTION
% OF THIS SOFTWARE ONLINE OR OFFLINE IS
% STRICTLY PROHIBITED.

%% Introduction and Organization
% This program was designed to temporally resolve temperature
measurements
% from high speed color camera images collected by PhantomVision
Research
% Cameras (notably the Miro M110). Previous applications
(PyCine_V2.py and
% CinveImageViewer_RGB_Extraction.m) have been utilized to create a
% calibration curve for the Miro M110 Cameran that can be used to
relate
% blackbody temperature to colors collected by the sensor on the
camera.

% This program wouldnt help or run with reasonable time without
input from
% Phil Guerieri and Rohit Jacob. Their help is greatly appreciated.

% The code below will be split up into the following sections:
% 1. Import of images using MatCine
% 2. Calibration Curve Information
% 3. Temperature Calculation
% 4. Maximum, Mean, and Median Temperature Measurements
% 5. Data Saving
% 6. Video Setup for Colormap Limits & Time

```

```

% 7. Temperature Only Video Creation
% 8. Side-by-side Original Color Video and Filtered Temperature
Video
% 9. Cleanup of Directories

clc
clear
close all
tic

% Adds all external functions to working path.
addpath(genpath(fileparts(which(mfilename))))

fprintf('Temperature Video Creator and MatCine\n');
fprintf('Owners: Dylan J. Kline, Michael R. Zachariah\n');
fprintf('=====\n');
fprintf('UNAUTHORIZED USE, DUPLICATION, AND/OR REDISTRIBUTION\nOF\nTHIS SOFTWARE ONLINE OR OFFLINE IS STRICTLY PROHIBITED.\n');
fprintf('=====\n');

%% 1. Import of Images using MatCine

% Pick .cine or .mat file using ui - multiple files can be selected
[files, path] = uigetfile({'*.cine;*.mat'}, 'Pick a Cine File');
% GUI selection of files
filename = fullfile(path, char(files));
% Extracts the filename and path
[pathstr,name,ext] = fileparts(filename);
% separates parts of filename and path

% Extraction of images using MatCine
if ext(end-3:end) == '.mat'
    load(filename,'raw_image_array')
% should import the data as a matlab file
else
    [header, bitmap, setup, raw_image_array] = MatCine(filename);
% runs MatCine program to extract information
end

% Details about video for later calculations
im_width = setup.ImWidth;
% extracts image width (column-wise)
im_height = setup.ImHeight;
% extracts image height (row-wise)
num_pixels = im_width * im_height;
% gets pixels per image
num_images = header.ImageCount;
% calculates number of images in file.

% Black level and saturation level values
filters.blacklevel = 64;
% Blacklevel as defined by the manufacturer
filters.blacklevelfactor = 1.5;
% Empirically chosen blacklevel based on experiments
filters.satlevel = 4000;
% Saturation level chosen empirically

% Time Setup
if ext(end-3:end) == '.cine'
    framerate = setup.Framerate16;
% framerate extraction based on the data from MatCine
else

```

```

    framerate = input('Framerate (pps): ');
% user-entered in the case where time is not included
end

if framerate>0
    if input('Was the video rerecorded and decimated? (y/n): ','s')
    == 'y' % some users needed to resave their videos at lower framerates
    and PCC was not recognizing that
        decimated = input('Decimated by: ');
% user-entered decimation
    else
        decimated = 1;
% if it wasn't decimated and they accidentally hit a number
    end
    time = (0:num_images-1)./framerate*decimated;
% calculates time based on framerate
else
    time = (0:num_images-1);
% Just puts in times if (for whatever reason) framerate was not
% saved. Rarely used.
end

%% 2. Calibration Curve Information

% Determines the calibration curve to match temperatures with.
% load('ratios_MiroM110_273-2343.mat') %
for calibration only
load('ratios_MiroM110_773-4773.mat'); %
recently calibrated 05/11/17
lambdacorrect = input('Lambda Correction (y/n): ','s');
if lambdacorrect == 'y' %
    load('ratios_MiroM110_773-4773_lambdacorrect.mat'); %
    correction for 1/lambda in values of epsilon
elseif lambdacorrect == 't' %
    load('ratios_MiroM110_773-5773_lambdacorrect.mat'); %
    high temperature hidden option, correction for 1/lambda in values of
    epsilon
% elseif lambdacorrect=='m' %
    removed 08/20/19 since we purchased the K2 Distamax
% load('ratios_MiroM110_773-4773_microscopelambdacorrect.mat'); %
    microscope hidden option with correction for 1/lambda.
end

% User-entered information about the camera lens setup.
fprintf('\nChoose lens for calibration factors\n') %
different lenses have different calibration factors.
calibchoice=input('Micro (1), Macro (2), or Wide Angle (3): '); %
user picked lens
if calibchoice == 1 %
    calibFactors = [1.0842 1.1932 1.2931]; %
    K2 distamax microscope lens tested at China Lake
elseif calibchoice == 2 %
    calibFactors = [0.952 0.888 0.847]; %
macro lens
elseif calibchoice == 3 %
    calibFactors = [1.007 0.966 0.972]; %
wide angle
else
    fprintf('\nLENS OPTION NOT RECOGNIZED.\n');
    return
end

%% 3. Temperature Calculation

```

```

filters.errorThresh=input('Error Threshold (0.1 default): '); %
Included this for storage of the error for later processing
if isempty(filters.errorThresh) == 1 %
Option to check and see if the string for error thresholding is empty
    filters.errorThresh = 0.1; %
Default to 0.1 if user defined error threshold is empty
end
Tdm = ones(1,3,length(ratios_summary)); %
initializing of Tdm
Tdm(1,(:,)) = ratios_summary(:,2:4)'; %
rearranging the calibrated ratios
Timage = nan(im_height,im_width,num_images); %
Initializing array for temperature information
TErrorStack = nan(im_height,im_width,num_images); % Data
for storage of error information

% This will be the size of the window to check for thresholding of
saturation and black levels.
se=strel('square',3);

% Set threshold for 0 in calculations (removes data that is
extraneous)
filters.zero = (raw_image_array < (filters.blacklevel *
filters.blacklevelfactor));
filters.zero = imdilate(filters.zero,se);

% Set threshold for saturated pixels to remove distortion in
calculations based on the demosaicing algorithm.
filters.sat = (raw_image_array > filters.satlevel);
filters.sat = imdilate(filters.sat,se);

% Sets threshold of ratios for calculations
% row1:minimum and row2:maximum ratio values for G/R, B/G, and B/R
respectively
filters.ratioThresh = [0.59 0.31 0.20;...
                      1.22 0.96 0.85];

% Tao was having high temperature materials and ratio thresholds need
to be
% changed to reflect changes in spectral curve
if lambdacorrect=='t' % t for tao
    filters.ratioThresh = [0.61 0.33 0.22;...
                          1.40 0.94 1.00];
end

% Haiyang and I were experimenting and found that if we basically
widened
% the ratio thresholding then we could still get reasonably good
data. This
% option essentially removes the error thresholding entirely (or at
least
% makes it so wide that it doesnt exist).
if lambdacorrect=='h' % for haiyang
    filters.ratioThresh = [0.51 0.23 0.12;...
                          1.50 1.04 1.10];
end

% Initializing average error matrix
TErrorAve = zeros(num_images,4);
TErrorAve(:,1) = time';

```

```

clearvars temp

wait = waitbar(i,sprintf('Matching temperatures...\nImage:
%i/%i',i,num_images));
for i=1:num_images

    waitbar(i/num_images,wait,sprintf('Matching
temperatures...\nImage: %i/%i',i,num_images));

    % Demosaics image
    demosaicIm = demosaic(raw_image_array(:, :, i), 'gbgr'); %
    demosaiced raw image array
    R = double(demosaicIm(:, :, 1)); % R pixel
    values for frame i
    G = double(demosaicIm(:, :, 2)); % G pixel
    values for frame i
    B = double(demosaicIm(:, :, 3)); % B pixel
    values for frame i

    % Sodium Test
    % R(90:110,115:135)=2.5*R(90:110,115:135);
    % G(90:110,115:135)=2.5*G(90:110,115:135);

    % Creates array with [pixel#, G/R calibrated ratio, B/G
    calibrated ratio, and B/R calibrated ratio.
    dmMat = [(1:num_pixels)'...
              reshape((G./R)/calibFactors(1),num_pixels,1)...
              reshape((B./G)/calibFactors(2),num_pixels,1)...
              reshape((B./R)/calibFactors(3),num_pixels,1)];

    % Establishes thresholds and reduces image down to pixels that
    fit threshold
    filters.ratio = reshape((dmMat(:,2) >
filters.ratioThresh(1,1)...
    & dmMat(:,3) >
filters.ratioThresh(1,2)...
    & dmMat(:,4) >
filters.ratioThresh(1,3)...
    & dmMat(:,2) <
filters.ratioThresh(2,1)...
    & dmMat(:,3) <
filters.ratioThresh(2,2)...
    & dmMat(:,4) <
filters.ratioThresh(2,3))...
, num_pixels,1);

    dmMat = dmMat(filters.ratio,:);
    % reduces matrix down to values that are within threshold
    TError = nan(im_height,im_width);
    % sets up error for images

    [TError(dmMat(:,1)),Ti] = min(squeeze(sum(abs(dmMat(:,2:4)-
Tdm),2))'); % takes the absolute difference between the ratios and
calculated values and sums them. after this it squeezes the matrix
into 2D and then seeks the position of the minimum values in each
column.

    T = nan(im_height,im_width);
    % creates empty array of NaN values.
    T(dmMat(:,1)) = ratios_summary(Ti,1);
    % places values for temperature based on position values found in
    lines above. Remainder of pixels in photo are NaN

```

```

    TErrorStack(:, :, i) = TError;
% Stores locations of errors
    Timage(:, :, i) = T;
% appends image to stack of images

    % creating a cell stack for error analysis later
    %Tcell=cell(im_height,im_width);
% creation of empty cell of the image for errors
    temp(:, 1)=dmMat(:, 1);
    if isempty(temp) ~= 1
        temp(:, 2:4) = (dmMat(:, 2:4)-ratios_summary(Ti, 2:4));
% getting all of the values in error for the accepted ones to this
point, stored as cells
        temp(:, 5) = sum(abs(temp(:, 2:4)), 2);
% creates a fifth column as the sum of all other columns for ratio
used for thresholding later
        temp(find(temp(:, 5)>filters.errorThresh), :) = [];
% thresholding of values above the error threshold
        TErrorAve(i, 2:4) = abs(mean((temp(:, 2:4)), 1));
% storing of average error in each ratio by frame.
    end
    clear temp

end

% Post processing to eliminate values above and below threshold for
calculation.
filters.nan = isnan(Timage);

close(wait)
toc
clearvars Ti temp Tdm dmMat T

%% 4. Maximum, Mean, and Median Temperature Measurements
% Creates an array that only contains the values that were
acceptable for
% analysis. Also, calculates light intensity.

filters.error = TErrorStack > filters.errorThresh; % Boolean matrix
stores where high error values are
filters.T_low = 773; % low temperature
threshold
filters.contiguous = 10; % minimum number
of contiguous pixels necessary to be used as part of mean/median
filters.lightThresh = 0.1; % amount of light
necessary before calculating mean/median

% making a normalized light intensity calculation
for i=1:num_images
    light_intensity(i) = sum(sum(sum(raw_image_array(:, :, i)))); %
Integrated amount of light hitting the sensor
end
light_intensity=light_intensity/max(light_intensity); %
Normalizes light intensity values

% This section is weird and has been added primarily for Haiyang and
% myself. Oftentimes we would have these films that would burn on a
glass
% slide and in the background of the image the plume would be read as
valid
% temperature measurements, when in reality you were looking at

```

```

% temperatures far from the glass. Since we are focused on things
that are
% close to the glass and those spots are usually pretty bright in
% comparison to far away from the glass, I added an option to make a
black
% and white overlay that will remove the "dark" background. Not sure
if
% this will be a long term solution or not.
if input(sprintf('\nSelect bright reaction front? (y/n): '), 's') ==
'y'
    [~,brightframe] = max(light_intensity);
    gain = 4096/max(max(raw_image_array(:, :, brightframe)));
    bw_thresh = input(sprintf('\tBinary threshold (0.25-0.4
recommended): '));
    for k = 1:num_images
        filters.front(:, :, k) =
            rgb2gray(uint8(double(demosaic(gain*raw_image_array(:, :, k), 'gbgr'))*2
55/4096))); % 1 means it is dark, 0 means it is bright
    end
    filters.front = rescale(filters.front);
    filters.front = filters.front > bw_thresh;
else
    filters.front = ones(im_height, im_width, num_images);
end

% creates an array of values for consideration in the calculations
below
% that sets all NaNs to 0 and uses the bool values calculated above
for
% data reduction.
filters.summary = double(Timage).*(1-filters.nan).*(1-
filters.zero).*(1-filters.sat).*(1-filters.error).*filters.front; %
copies and thresholds Timage to create an acceptable summary
Tmean(:, 1) = time';
% initializing time in Tmean
Tmedian(:, 1) = time';
% initializing time in Tmedian
Tmax(:, 1) = time';
% initializing time in Tmax

% Average Temperature
wait = waitbar(i, sprintf('Statistical analysis...\nImage:
%i/%i', i, num_images));
for i = 1:size(Timage, 3)

    waitbar(i/num_images, wait, sprintf('Statistical
analysis...\nImage: %i/%i', i, num_images));

    clearvars temp

    temp =
uint16(bwareaopen(imbinarize(filters.summary(:, :, i)), filters.contiguo
us)).*uint16(Timage(:, :, i)); % finds contiguous areas to make
measurements with
    t2(i) = nnz(temp);

    % after ignition wants it to exclude points after it gets dark
    if light_intensity(i) > filters.lightThresh && t2(i) > 0 % count
number of nonzero elements, needs to be greater than 0.2% to be
considered for temperature calculation
        Tmean(i, 2) = mean(mean(temp(temp > 0))); %
determine mean temperature
    end
end

```



```

        Tmax(i,2) = max(max(temp(temp>0))); %
determine max temperature
        Tmedian(i,2) = median(median(temp(temp>0))); %
determine median temperature
    end

end

% Remove zeros from data stored above (for points excluded)
Tmean(all(Tmean(:,2)<=filters.T_low,2),2) = NaN;
Tmedian(all(Tmedian(:,2)<=filters.T_low,2),2) = NaN;
Tmax(all(Tmax(:,2)<=filters.T_low,2),2) = NaN;
Tmean(all(isnan(Tmean(:,2))==1,2),2) = NaN;
Tmedian(all(isnan(Tmedian(:,2))==1,2),2) = NaN;
Tmax(all(isnan(Tmax(:,2))==1,2),2) = NaN;
close(wait)

% Figure creation
fig=figure('Color',[1 1 1]);
set(fig, 'defaultAxesColorOrder',[0 0 0]; [1 0 0])
hold on

yyaxis left
scatter(1000*Tmean(:,1),Tmean(:,2),65,'o','MarkerFaceColor',
[0.4660 0.6740 0.1880],'MarkerEdgeColor','k')
scatter(1000*Tmedian(:,1),Tmedian(:,2),65,'o','MarkerFaceColor',
[0.8500 0.3250 0.0980],'MarkerEdgeColor','k')
xlabel('time (ms)')
ylabel('Temperature (K)')
xlim([1000*min(time) 1000*max(time)])

yyaxis right
plot(1000*time,light_intensity,'r','LineWidth',1.75)
ylabel('Normalized Light Intensity')
ylim([0 1])

title(sprintf('Temperature Profiles for
%s',name),'interpreter','None')
legend('Mean','Median','Light Intensity')
grid('on')

ax=gca;
ax.FontSize=14;
ax.LabelFontSizeMultiplier = 1.25;
ax.TitleFontSizeMultiplier = 1.35;
hold off

savefig(sprintf('%s_maxmean.fig',horzcat(path,name)))
pause(4.5) % you know, so your eyes can see it
close all

%% 5. Data Saving
Timage = uint16(Timage);
%TErrorStack=int16(TErrorStack);
fprintf('\nSaving data... ');
save(char(sprintf('%s.mat',horzcat(path,name))), 'num_images','files',
'Timage','filters','raw_image_array','R','G','B','calibFactors',
'filename',
'TErrorStack','Tmedian','Tmax','Tmean','time','TErrorAve','framerate',
'light_intensity','setup','name','path');
fprintf('Done saving :)\n');
close all

```

```

clearvars im_construct filters.summary t2

%% 6. Video Setup for Colormap Limits & Time
% User chosen colormap limits

fprintf('\nChoose colormap limits or define your own\n');
vidT.cmapchoice = input('Auto (1) or Custom (2): ');
if vidT.cmapchoice == 1
    vidT.cmaplim = [0.85*nanmean(Tmedian(:,2))
1.15*nanmean(Tmedian(:,2))]; % just to get a rough idea.
else
    vidT.cmaplim = input(sprintf('\tEnter colormap limits as array:
')); % in case you are a special snowflake.
end

% Displays information about image for user to make decision about
video creation
fprintf('\nRecorded video information:\n');
fprintf('\tNumber of Frames: %d\n', num_images);
% displays number of frames
fprintf('\tRecorded length of video: %fs\n', max(time));
% displays the length of the video based on framerate
fprintf('\tTime per frame: %fs\n', time(2));
% displays incremented time per frame in the original recorded video

% Users can choose to make their video length based on framerate or
time.
% Nominally, users would pick the framerate and then choose
something
% between 30-60 frames/s, but the choice is there for other options.
vid.framerateortime = input(sprintf('\nDefine video length based on
time (1) or framerate (2)? : '));
if vid.framerateortime==1
    vid.vidtime = input(sprintf('\tDesired length of video (s): '));
% prompts user for input regarding the length of the video to be
generated
    vid.imstep = input(sprintf('\tSteps between frames (1 for no
skipped frames): '); % prompts user to enter the amount of images to
skip between frames
    vid.vidtime = vid.vidtime * vid.imstep;
% Necessary to correct time for encoding below
else
    vid.enteredframerate = input(sprintf('\tDesired framerate
(frames/s): '); % prompts user for entered desired
framerate, useful for slides with multiple videos to compare speed
    vid.imstep = input(sprintf('\tSteps between frames (1 for no
skipped frames): '); % prompts user to enter the amount of images
to skip between frames
    vid.vidtime = num_images/vid.imstep/vid.enteredframerate;
    fprintf('\tVideo Length: %fs\n', vid.vidtime);
    vid.vidtime = vid.vidtime * vid.imstep;
% Necessary to correct time for encoding below
end

% This was more or less implemented because some people would get
dark
% spots that were not really explained well. In previous iterations,
% this was simply left black because they were typically dark or
% unburned regions, but now that people use some molecular species
% which absorb a lot of light (like iodine), it is important to add
an
% option to visualize that the camera is indeed capturing the whole

```

```

% plume but is not able to assign any reasonable temperature to it
% because of the ratios being, for lack of a better word, "wack."
if input(sprintf('\nShow gray (out of bounds) pixels? (y/n): '), 's')== 'y'
    vidT.rgbRatio = [192 192 192]; % should be pretty gray
else
    vidT.rgbRatio = [0 0 0]; % if people don't want to see the
out of bounds pixels
end

% Color of high error pixels in RGB
vidT.rgbError = [101, 66, 66]; % about copper colored

% In case video was recorded in a rotated manner, generally not a
problem
if input(sprintf('\nIs the image rotated? (y/n): '), 's') == 'y'
    vid.rotk = input(sprintf('\tNumber of 90-degree clockwise
rotations to apply to image: ')); %number of image rotations
else
    vid.rotk = 0; % no image rotations
end

% Gain adjustment to make videos brighter
% This auto gain routine can be used to make the brightest pixel
white. If
% a user chooses something other than auto gain, then it will ask for
a
% value and display the calculated auto gain. Some users may choose
to run
% this section using the custom gain option and then choose the same
value,
% though typically 1.5-2x the auto gain value does a very good job.
[~,brightframe]=max(light_intensity);
vid.gain=4096/max(max(raw_image_array(:,:,brightframe)));
if input('\nVideo gain auto (1) or custom (2)? : ')==1
    fprintf('\tGain: %d\n',vid.gain);
else
    fprintf('\tAuto gain: %d\n',vid.gain);
    vid.gain=input(sprintf('\tDesired gain: '));
end

%% 7. Temperature Only Video Creation
% Creates an mp4 video only showing the false color temperature
images.
% White denotes where oversaturation occurred, copper colored pixels
are
% high in error, and gray colored pixels are outside of the ratio
% thresholding bounds. All of the video parameters are determined
using
% the prompts in the section above.

ogvid= input(sprintf('\nWould you like a temperature only video?
(y/n): '), 's');

if ogvid=='y'
    v = VideoWriter(sprintf('%s.mp4',horzcat(path,name)), 'MPEG-4');
% creates video to be written
    v.FrameRate = round(num_images/vid.vidtime);
% autocalculated based on the desired length and image steps
    v.Quality = 100;
% 75 is default, increased to look pretty
    open(v)

```

```

figure('units','pixels','position',[0 0 1.6*im_width
1.2*im_height]) % figure auto adjusted for original size of video

ax1 = subplot(1,1,1);

imSize = [im_height im_width];

% rotates images
bg.zeroIm = rot90(cat(3, zeros(imSize), zeros(imSize),
zeros(imSize)),vid.rotk);
bg.satIm = rot90(cat(3, ones(imSize), ones(imSize),
ones(imSize)),vid.rotk);
bg.ratioIm = rot90(cat(3, ones(imSize), ones(imSize),
ones(imSize)).*0.0,vid.rotk);
bg.errIm = rot90(cat(3, vidT.rgbError(1)*ones(imSize)./255,
vidT.rgbError(2)*ones(imSize)./255,
vidT.rgbError(3)*ones(imSize)./255),vid.rotk);

% creates an error bool for display later
filters.error = TErrorStack>filters.errorThresh;

% Iterative display of images to be written to video
for k=1:vid.imstep:num_images

%Copy the Timage displayed to axes3
bg.zero = imagesc(bg.zeroIm,'Parent',ax1);
hold(ax1,'on')
title(ax1,sprintf('%s',name),'interpreter','None')
bg.sat = image(bg.satIm,'Parent',ax1);
bg.ratio = image(bg.ratioIm,'Parent',ax1);
bg.err = image(bg.errIm,'Parent',ax1);
Tim3 = imagesc(ax1,rot90(Timage(:,:,k),vid.rotk));

%contour(bwareaopen(im2bw(filters.summary(:,:,k)),35),1,'Color',[255
20 147]./255);
set(Tim3, 'AlphaData',
not(rot90(filters.nan(:,:,k),vid.rotk) |
rot90(filters.zero(:,:,k),vid.rotk) |
rot90(filters.sat(:,:,k),vid.rotk) |
rot90(filters.error(:,:,k),vid.rotk) |
rot90(~filters.front(:,:,k),vid.rotk)))
set(bg.err, 'AlphaData',
not(rot90(filters.nan(:,:,k),vid.rotk)|rot90(filters.zero(:,:,k),vid.
rotk)|rot90(filters.sat(:,:,k),vid.rotk) |
rot90(~filters.front(:,:,k),vid.rotk)));
set(bg.ratio, 'AlphaData',
not(rot90(filters.zero(:,:,k),vid.rotk)|rot90(filters.sat(:,:,k),vid.
rotk) | rot90(~filters.front(:,:,k),vid.rotk)));
set(bg.sat, 'AlphaData',
not(rot90(filters.zero(:,:,k),vid.rotk) |
rot90(~filters.front(:,:,k),vid.rotk)));
hold(ax1,'off')

daspect(ax1,[1 1 1])
if k==1
origAx1Size = ax1.Position-[0.075 0 0 0];
end

vidT.myColorMap=jet(256);
colormap(vidT.myColorMap);
caxis(ax1,vidT.cmaplim)

```

```

vidT.c = colorbar('eastoutside');
set(vidT.c, 'fontsize', 14)
set(vidT.c, 'fontweight', 'bold')
vidT.c.Label.String = 'Temperature (K)';
set(ax1, 'Position', origAx1Size);

% creates text boxes displaying the framerate, the frame
number, and a timestep
if framerate>0
    if k==1
        AxesHandle=findobj(gcf, 'Type', 'axes');

timelab=annotation('textbox',plotboxpos(ax1), 'string', sprintf('%.2fms
',time(k)*1000), 'FitBoxToText', 'on', 'Color', 'w', 'FontSize', 12, 'Backgr
oundColor', 'none', 'EdgeColor', 'none');
    else
        timelab.String=sprintf('%.2fms',time(k)*1000);
    end
end

f=getframe(figure(1));
writeVideo(v,f)

end
close(v)

% finishing of this section of code will be indicated by closing
of the figure.
close all

end

%% 8. Side-by-side Original Color Video and Filtered Temperature
Video
% Creates a new video where white shows where oversaturation
occured,
% copper is points with errors in the thresholding calculations
% (temperatures may not be correct) and gray are where there is data
there
% but we are not able to use it reasonably. Images with large
% concentrations of any color should be heavily speculated.

% User option to create a side-by-side T video
filtervid= input(sprintf('\nWould you like a side-by-side T video?
(y/n): '), 's');

if filtervid=='y'

    %orient=input('Horizontal (1) or Vertical Alignment (2) of
Videos: ');
    orient = 1; % restricting people to just do horizontal videos to
make life easy
    i=0;
    wait = waitbar(i, sprintf('Processing image %i/%i', i, num_images));
    v =
VideoWriter(sprintf('%s_filtered.mp4', horzcat(path, name)), 'MPEG-4');
    v.FrameRate = round(num_images/vid.vidtime);
    v.Quality = 100;
    open(v)

    vid.rotk = 0; %number of image rotations

```

```

    if mod(vid.rotk,2)~=0
        fig_height=im_width;
        fig_width=im_height;
    else
        fig_height=im_height;
        fig_width=im_width;
    end

    aspect=[fig_width fig_height 1];

    % dynamic display of figure based on dimensions of a screen. how
    about that for cool
    figdim = round(1280/2*fig_height/fig_width); % ratio of height to
    width
    fig = figure('units','pixels','position', [0 0
    2*figdim*fig_width/fig_height figdim+100],'Color',[1 1 1]);
    ax1 = axes('Units', 'pixels', 'Position', [0 100
    figdim*fig_width/fig_height figdim]);
    ax2 = axes('Units', 'pixels', 'Position',
    [figdim*fig_width/fig_height 100 figdim*fig_width/fig_height
    figdim]);

    %filters.errorThresh = 0.1; %threshold for maximum error in
    sum(abs(calc(ratios)-theory(ratios)))
    imSize = [im_height im_width];

    % Pixels show up a copperish color when they are considered to be
    too
    % high in error based on the error thresholding chosen by the
    user.
    vidT.rgbError=[101, 66, 66]; % about copper colored

    % rotates images
    bg.zeroIm = rot90(cat(3, zeros(imSize), zeros(imSize),
    zeros(imSize)),vid.rotk);
    bg.satIm = rot90(cat(3, ones(imSize), ones(imSize),
    ones(imSize)),vid.rotk);
    %bg.ratioIm = rot90(cat(3, ones(imSize), ones(imSize),
    ones(imSize)).*0.0,vid.rotk);
    bg.ratioIm = rot90(cat(3, vidT.rgbRatio(1)*ones(imSize)./255,
    vidT.rgbRatio(2)*ones(imSize)./255,
    vidT.rgbRatio(3)*ones(imSize)./255),vid.rotk);
    bg.errIm = rot90(cat(3, vidT.rgbError(1)*ones(imSize)./255,
    vidT.rgbError(2)*ones(imSize)./255,
    vidT.rgbError(3)*ones(imSize)./255),vid.rotk);

    % creates an error bool for display later
    filters.error = TErrorStack>filters.errorThresh;

    for k=1:vid.imstep:num_images

        waitbar(k/num_images,wait,sprintf('Processing image
        %i/%i',k,num_images)); % update waitbar

        % Displays original color image
        temp = demosaic(vid.gain*raw_image_array(:,:,k),'gbrg'); %
        temporary array with gain and demosaiced raw image array
        temp(:,:,1) = setup.WBGain1.R*temp(:,:,1); %
        red channel gain to make colors look better
        temp(:,:,3) = setup.WBGain1.B*temp(:,:,3); %
        blue channel gain to make colors look better
    
```

```

        temp = rot90(temp,vid.rotk); %
    apply rotations if appropriate
        image(double(temp)/4096,'Parent',ax1) %
    show image on left panel
        hold(ax1,'on')
        daspect(ax1,[1 1 1]) %
    change pixel data aspect ratio so the video looks normal and not
    distorted
        hold(ax1,'off')

        %Copy the Timage displayed to axes3
        bg.zero = imagesc(bg.zeroIm,'Parent',ax2);
        hold(ax2,'on')
        bg.sat = image(bg.satIm,'Parent',ax2); %
    white panel that is placed on bottom
        bg.ratio = image(bg.ratioIm,'Parent',ax2); %
    gray panel that is placed on bottom
        bg.err = image(bg.errIm,'Parent',ax2); %
    copper panel that is placed on bottom
        Tim3 = imagesc(ax2,rot90(Timage(:,:,k),vid.rotk)); %
    Temperature image
        %set(Tim3, 'AlphaData',
    not(rot90(filters.nan(:,:,k),vid.rotk) |
    rot90(filters.zero(:,:,k),vid.rotk) |
    rot90(filters.sat(:,:,k),vid.rotk) |
    rot90(filters.error(:,:,k),vid.rotk)))
        set(Tim3, 'AlphaData',
    not(rot90(filters.nan(:,:,k),vid.rotk) |
    rot90(filters.zero(:,:,k),vid.rotk) |
    rot90(filters.sat(:,:,k),vid.rotk) |
    rot90(filters.error(:,:,k),vid.rotk) |
    rot90(~filters.front(:,:,k),vid.rotk)))
        set(bg.err, 'AlphaData',
    not(rot90(filters.nan(:,:,k),vid.rotk)|rot90(filters.zero(:,:,k),vid.
    rotk)|rot90(filters.sat(:,:,k),vid.rotk) |
    rot90(~filters.front(:,:,k),vid.rotk)));
        set(bg.ratio, 'AlphaData',
    not(rot90(filters.zero(:,:,k),vid.rotk)|rot90(filters.sat(:,:,k),vid.
    rotk) | rot90(~filters.front(:,:,k),vid.rotk)));
        set(bg.sat, 'AlphaData',
    not(rot90(filters.zero(:,:,k),vid.rotk) |
    rot90(~filters.front(:,:,k),vid.rotk)));

        daspect(ax2,[1 1 1]) %
    change pixel data aspect ratio so the video looks normal and not
    distorted
        origAx2Size= ax2.Position; %
    rescaling the axis so it fits appropriately

        vidT.myColorMap = jet(256); %
    creating a colormap for the dataset
        colormap(vidT.myColorMap); %
    false color image with color map
        caxis(ax2,vidT.cmaplim) %
    show colormap axis
        hold(ax2,'off')

        if framerate>0
            if k==1
                timelab = annotation('textbox',
    'units','pixels','position',[15, 50, figdim*fig_width/fig_height,
    40], 'string', sprintf('t: %.3f

```

```

ms',time(k)*1000),'FitBoxToText','on','FontSize',30,'BackgroundColor'
,'none','EdgeColor','none');
    frameno = annotation('textbox',
'units','pixels','position',[15, 10, figdim*fig_width/fig_height,
40], 'string', sprintf('frame:
%d',k),'FitBoxToText','on','FontSize',30,'BackgroundColor','none','Ed
geColor','none');
    %medianlab = annotation('textbox',
'units','pixels','position',
plotboxpos(ax2),'string',sprintf('Median: %d
K',Tmedian(k,2)),'FitBoxToText','on','Color','w','FontSize',25,'Backg
roundColor','none','EdgeColor','none');
    %titlelab = annotation('textbox',
'units','pixels','position', [0.5*figdim*fig_width/fig_height, 50,
0.25*figdim*fig_width/fig_height,
40], 'string', sprintf('%s',name),'FitBoxToText','on','FontSize',15,'Ba
ckgroundColor','none','EdgeColor','none');
    else
        timelab.String = sprintf('t: %.3f ms',time(k)*1000);
        frameno.String = sprintf('frame: %d',k);
        %medianlab.String = sprintf('Median: %d
K',Tmedian(k,2));
    end

    % does all the colorbar business
    vidT.c=colorbar('southoutside');
    set(vidT.c,'fontsize',14)
    set(vidT.c,'fontweight','bold')
    vidT.c.Label.String = 'Temperature (K)';
    set(ax2, 'Position', origAx2Size);

set(vidT.c,'units','pixels','Position',[1.025*figdim*fig_width/fig_he
ight 50 0.95*figdim*fig_width/fig_height 20])

    end

    f=getframe(figure(1)); % get frame
    writeVideo(v,f)        % write frame

    end

close(v)
close(wait)
end

close all

%% 9. Cleanup of Directories
% This will create a subfolder with format YYYYMMDD 'Filename' and
store
% all of the files inside of it
cleanup = input(sprintf('\nWould you like to clean up directories?
(y/n): '), 's');
if cleanup=='y'
    basename = filename(1:end-size(ext,2));
    foldername=fullfile(path, sprintf('%s
%s',string(datetime('now','Format','yyyyMMdd')), string(name)));
    mkdir(foldername);

    movefile(sprintf('%s.mat',basename),foldername); %
moves mat file

```



```

        movefile(sprintf('%s_maxmean.fig',basename),foldername); %
moves mean/median figure file

        if ogvid=='y'
            movefile(sprintf('%s.mp4',basename),foldername); %
moves temperature only video
        end

        if filtervid=='y'
            movefile(sprintf('%s_filtered.mp4',basename),foldername); %
moves filtered video file
        end

        if exist(sprintf('%s.cine',basename),'file')==2
            fclose('all');
            movefile(sprintf('%s.cine',basename),foldername); %
moves cine file
        end
end

```

## Appendix E: Experimental description for high-speed microscopy

### ***E.1. Introduction***

As part of the evolution of the research performed over the course of this dissertation, the color camera pyrometer has been adapted for use as a high-speed microscope with temperature measurement capabilities using the pyrometry code. This idea was originally conceived with Dr. Rohit Jacob during his time in the Zachariah Group and was recommended work as part of his dissertation, but was first implemented as part of a publication by Wang & Kline et al.<sup>166</sup> Calibration information for the long working distance microscope lens (InfinityOptics) was later detailed in Kline & Alibay et al.<sup>238</sup>

Use of this high-speed microscopy apparatus enables high-resolution images to be captured of the combustion process *in-operando*, giving researchers the ability to witness different propagation regimes, evaluate heat transfer mechanisms, and estimate the reaction front length of combusting materials. Pyrometry measurements can be performed using the pyroCam code detailed in this dissertation using methods discussed in Jacob & Kline et al.<sup>73</sup> Later section in this dissertation also provide user manuals for codes developed to extract the reaction front length and estimate local velocities along the reaction front as first detailed in Kline et al.<sup>174</sup> This appendix will serve as a simple procedural manual for performing high-speed microscopy experiments as they have been described in these publications.

### ***E.2. Descriptions of available cameras/lenses***

The high-speed microscopy apparatus relies on the use of the high-speed cameras that are available for use in the Zachariah Group, however, the Vision Research Phantom VEO710L camera is the preferred camera for this experiment since it can achieve higher framerates and has been specifically calibrated for with the microscope lenses in the lab. The optics used in the microscopy apparatus have undergone two major iterations – (1) a Nikon Macro lens looking into a Nikon 40x microscope objective<sup>166</sup> and (2) an InfinityOptics K2 DistaMax coupled to a CF-4 objective and NTX tube.<sup>238</sup> Resolution details for both setups are provided below, however, at the time of writing this dissertation only the long working distance microscope is used since it has a longer working distance (reduces risk of lens damage during experiments) and has a better depth of field despite it having a slightly lower resolution than the other setup.

**Table E-1:** Microscope lens assembly options for use as a high-speed microscope/pyrometer. Reported resolution is for use with Vision Research Phantom VEO710L.

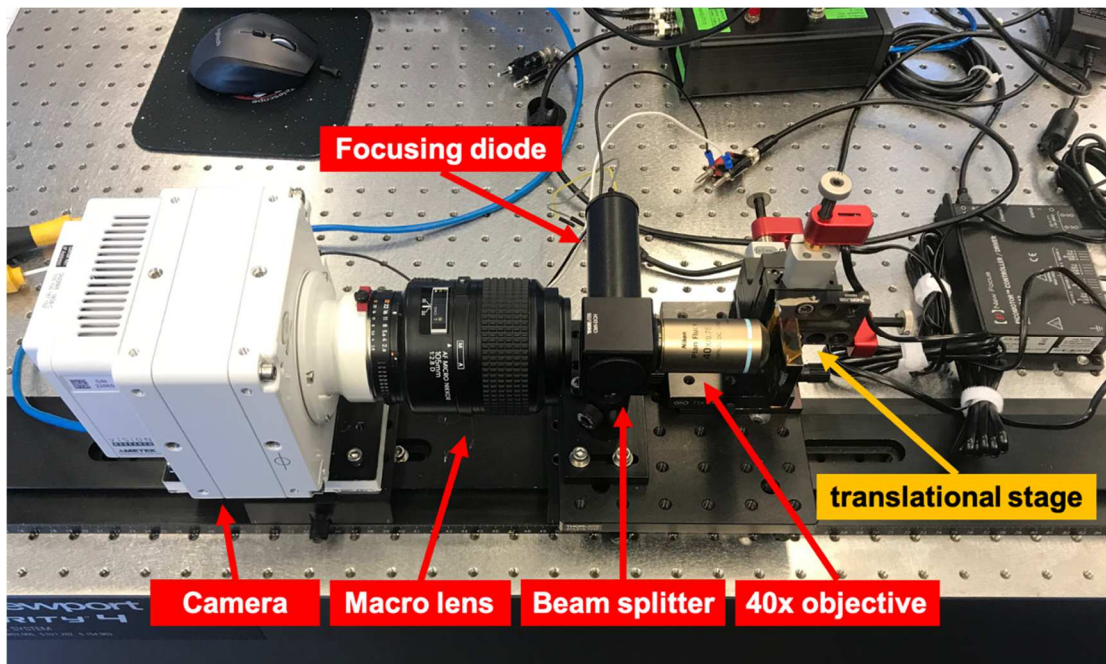
<b>Lens Assembly</b>	<b>Resolution</b>	<b>Working Distance</b>	<b>Depth of Field</b>
Nikon AF Micro Nikkor 105mm 1:2.8D macro lens → Nikon CFI Plan Fluor 40x microscope objective <sup>166</sup>	1 $\mu\text{m}/\text{px}$	0.66 mm	$\sim 1 \mu\text{m}$
InfinityOptics K2 DistaMax with CF-4 objective and NTX tube <sup>238</sup>	1.68 $\mu\text{m}/\text{px}$	54 mm	$\sim 10 \mu\text{m}$

### ***E.3. Procedures for performing high-speed microscopy***

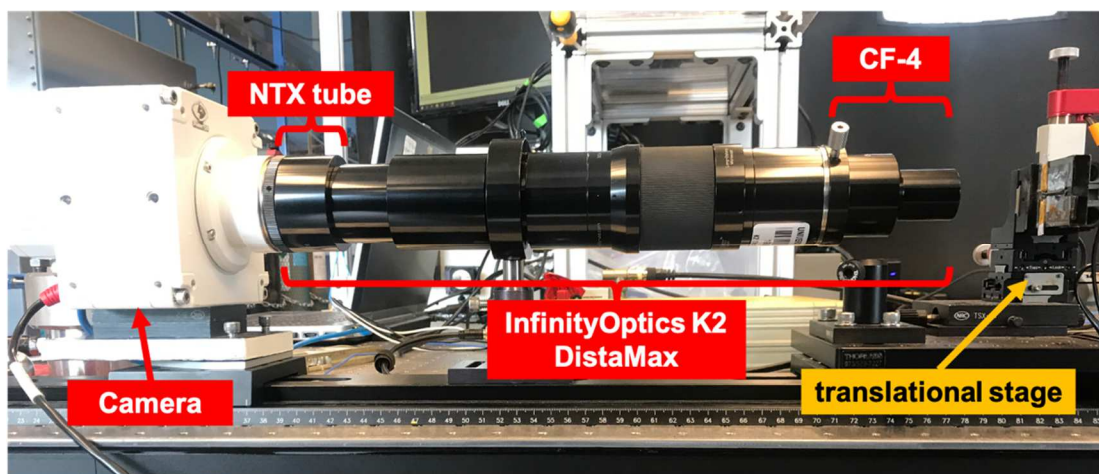
Operation of the high-speed microscope apparatus is nearly identical to operation of the high-speed color camera pyrometer (previously discussed in Appendix

A). The procedure is reproduced here for completeness with few changes made in consideration of the microscope.

To perform high-speed microscopy on combusting materials, it is recommended to trigger a heating wire (or other input power source) simultaneously with the camera triggering (discussed below) using a standardized TTL signal from any of the available trigger boxes in the lab (see Appendix E or Appendix F for sample TTL output boxes). The most commonly used setups of the high-speed microscopy experiment can be seen in Figure E-1 and Figure E-2. Detailed use of the Phantom Camera Control software is left to the user since documentation by Vision Research is readily available and provides good detail on operation.



**Figure E-1:** Example high-speed microscopy setup for Vision Research Phantom VEO710L equipped with a Nikon macro lens and Nikon Plan Fluor 40x microscope objective.



**Figure E-2:** Example high-speed microscopy setup for Vision Research Phantom VEO710L equipped with the InfinityOptics K2 DistaMax lens, CF-4 objective, and NTX tube.

1. Set up the camera on a stabilized surface (e.g. optical table, tripod) with the desired lens and connect to power and ethernet.
2. On a windows computer, start the Phantom Camera Control (PCC) software and connect to the camera.  
  
\*If the camera does not automatically connect, close PCC, change the network settings on the computer and set the IP address to 100.100.100.1 and the subnet mask to 255.0.0.0, then restart PCC.
3. If using a TTL triggering option on the camera, connect the Trigger input on the camera to the signaling device that will send the TTL signal.
4. Mount samples on a 3-axis translational stage (Newport) as either free-standing or printed on a glass microscope coverslip. Samples printed on coverslips and mounted with the coverslip bottom facing the lens are preferred since they improve the longevity of the life for the microscope lenses and gaseous side products will not obscure the view of the experiment. Use of 0.17 mm thick coverslips is required for experiments using the Nikon 40x objective.

5. Click “Low light” in PCC and focus on the desired object manually using window cropping options in PCC and translational stage.

When using the Nikon 40x objective assembly, use the red diode in the beam splitter as a light source to focus. If using the T-Jump Nano, a microscope code has been written that will power on the diode by turning on the switch on the box if the BNC output is connected to the diode.

When using the InfinityOptics microscope, it is recommended to use a flashlight held at an oblique angle to reflect light off of the sample surface for focusing.

Once focused, disable “Low light” option.

6. Set the desired framerate and exposure for the video to be taken.
7. Click CSR to save a closed shutter reference at the recording settings.
8. Click “Capture” in PCC.
9. Run experiment.
10. Review video in PCC by cropping the length of the video.
11. Save .cine file, make sure that the check box next to 16 bpp is enabled. If a video needs to be decimated, be sure to write down by which factor the video is decimated.
12. Repeat steps 4-10 for each experiment to be run.

13. Process videos for temperature measurements using the pyrometry codes as detailed in the software manual.
14. Process videos for reaction front length and local velocity measurements using the pyrometry and reaction front extractions codes as detailed in their respective software manuals.

## Appendix F: MATLAB script for extracting flame front surface area using high-speed microscopy

This appendix will serve as the user manual for the flame front extraction code as has been employed in the research described in 0 (among other articles that were published). The code was primarily designed as a use for high-speed microscopy experiments that are described in Chapter 4, 0, and 0. Throughout the manual, text in `this font` will represent variables or expressions that are used in a MATLAB command.

The manual below is broken up into different sections on (1) running the flame front extraction code, (2) input and output information for the program, (3) detailed variable listing and descriptions, (4) generalized code structure, (5) a summary, (6) version history notes if applicable, and (7) the raw MATLAB code.

### ***F.1. Running flame front extraction code***

The executable MATLAB script can be found in the Zachariah Group Drive as Volumes/users/Kline/Codes/Flame-Front-Extraction/kline\_microscope\_flamefront10.m. An online version has also been stored with version control and is available by reasonable request. The code has numerous dependencies on installed packages for MATLAB and those that have been written by the author and stored in Volumes/users/Kline/Codes/Flame-Front-Extraction/.

**Table F-1:** Dependencies for flame front extraction code.

Name	Description
MATLAB Image Processing Toolbox	General use toolbox which contains required functions for image processing related purposes. For this project in particular, regionprops is required.



MatCine	Used to extract raw image array from .cine file formats.
Boxcount	Used for fractal analysis when necessary, stored in external functions folder or available online at <a href="https://www.mathworks.com/matlabcentral/fileexchange/13063-boxcount">https://www.mathworks.com/matlabcentral/fileexchange/13063-boxcount</a> .

The opening lines of `kline_microscope_flamefront10.m` attempts to automatically add the required external functions based on the computer type of the user, but if this fails it can always be done manually by right-clicking on the External Functions folder and choosing Add to Path>Selected Folder and Subfolders.

To run `kline_microscope_flamefront10.m`, open the folder in the script viewer in MATLAB and click Run in the Editor Ribbon. The program will execute automatically from the first written line. Alternatively, users can choose to Run Section in the script viewing window since the code has been split up by section.

## ***F.2. Description of input and output***

The following parameters are used as input and outputs for the flame front extraction code and the prompts with their expected values are described below. Users will typically have to experimentally determine values for the different options if they find that the initial results are not satisfactory (i.e. poor fitting to the flame front).

### ***Input***

**Table F-2:** Input options and prompts for flame front extraction code with associated descriptions.

<b>Prompt</b>	<b>Description</b>
Pick Cine File (GUI)	Users are asked to navigate to the desired .cine file that they would like to be analyzed using the graphical user interface of their machine. The program will automatically save the path that the user has navigated to for future use in saving and organizing files.

	<p>Users may also choose to pick a .mat output file and the program will execute normally, though this feature is rarely used and it would be better to simply pick the .cine file.</p> <p>Variables stored:  files, path, filename, pathstr, name, ext, header, bitmap, setup, raw_image_array, im_height, im_width, pixNumel, num_images, im_construct</p>
Black level factor  Default (1) or other (2) :	<p>User option to use a custom black level factor for predetermination of pixels that are to be considered “black.” The camera black level is currently 64, so a black level factor of 1.5 would change the level to 96, thus all pixels with a 12-bit value &lt;96 would be automatically set to 0.</p> <p>Input 1 or 2.  If 2 is selected, the user must enter a new value.  If a number other than 1 or 2 is selected, users will also be prompted to enter a new value.</p> <p>Variables stored:  blackLevelFactor</p>
Framerate (pps) :	<p>If a user picked a .mat file without the time information, users will need to input the framerate. When loading a .cine file, this prompt is not shown since the framerate is automatically extracted from the header and setup files produced by MatCine.</p> <p>Input must be a number.</p> <p>Variables stored:  framerate</p>
Size calibration 1um/px (1) or 1.7um/px (2) :	<p>User option for size calibration of the pixels that will be used as a factor later when determining the total reaction front length and local velocities. The Nikon 40x microscope lens would be used for 1 <math>\mu\text{m}/\text{px}</math> sizes (old videos), but most videos now would be made using the InfinityOptics K2 Distamax lens which is currently limited to <math>\sim 1.7 \mu\text{m}/\text{px}</math>.</p> <p>Input options are 1 or 2.  If a number other than 1 or 2 is selected, the value will be set to <math>1.7 \mu\text{m}/\text{px}</math>.</p>

	<p>Variables stored: pxSize</p>
<p>Video gain Gain auto (1) or gain max (2) :</p>	<p>User option for aesthetics to make the rendered videos have a custom gain. Users can choose to use the auto gain which is calculated by saturating the brightest pixels in the brightest frame of the video or can simply set the gain to be 10 by choosing the maximum gain option.</p> <p>Input options are 1 or 2. If 2 is selected, the user must enter a new value. If a number other than 1 or 2 is selected, users will also be prompted to enter a new value.</p> <p>Variables stored: gain</p>
<p>Retain only longest chain? (y/n) :</p>	<p>User option to make the reaction front length only based upon the longest continuous chain of pixels measured in the video.</p> <p>Input options are y or n. If y is selected, only the longest chain will be shown in the videos and used for calculations. If n is selected, all non-continuous chains will also be under consideration for reaction front length. If something other than y or n is selected, the program will execute as if n was chosen.</p> <p>Variables stored: long_chain</p>
<p>Use entire frame as normalization length? (y/n) :</p>	<p>User option to change normalization basis for reaction front length. If the user chooses to not use the entire window as the normalization length, then the range of y values in each frame will be used as the normalization length. This feature is implemented, but rarely (if ever) used.</p> <p>Input options are y or n. If y is selected, the entire window frame will be used as the normalization length. If n is selected, the range of y values in the active frame will be used as the normalization length. If something other than y or n is selected, the program will execute as if y was chosen.</p> <p>Variables stored:</p>

	norm_y
Display vectors for local velocity on plot? (y/n):	<p>User option to display the vectors for the local velocity when generating the video to show the reaction front perimeter.</p> <p>Input options are y or n.  If y is selected, blue lines will be shown displaying the vectors between points in the reaction front in one frame to the next frame when rendering the video.  If something other than y is selected, the program will execute as if n was chosen.</p> <p>Variables stored:  plot_vectors</p>
Perform fractal analysis? (y/n):	<p>User option to perform fractal analysis. This region has been in experimentation and is rarely run but is left here for later refinement if necessary.</p> <p>Input options are y or n.  If y is selected, fractal analysis will be performed.  If n is selected, the code is skipped.  If something other than y or n is selected, the program will execute as if n was chosen.</p>
Create velocity histogram? (y/n):	<p>User option to create a local velocity histogram as calculated using the nearest-neighbor method.</p> <p>Input options are y or n.  If y is selected, a velocity histogram will be made.  If n is selected, the code section is skipped.  If something other than y or n is selected, the program will execute as if n was chosen.</p> <p>Files generated:  &lt;filename&gt;_veloc.fig</p>
Start frame for velocity analysis:	<p>Starting frame used for the velocity histogram analysis (if y was chosen above). Prompt not shown if n was selected above.</p> <p>Input must be an integer.</p> <p>Variables stored:  frame_start</p>

End frame for velocity analysis:	<p>Ending frame used for the velocity histogram analysis (if y was chosen above). Prompt not shown if n was selected above.</p> <p>Input must be an integer.</p> <p>Variables stored: frame_end</p>
X-axis in cm/s (1) or m/s (2):	<p>Option for changing the x-axis scale in the generated velocity histogram. Prompt not shown if n was selected above.</p> <p>Input options are 1 or 2. If a number other than 1 or 2 is selected, the program will execute as if 2 was chosen.</p> <p>Variables stored: cms_ms</p>
Would you like to clean up directories? (y/n):	<p>User option to create a new folder with all of the generated files in the newly created folder. Folder will have name format of yyyyMMdd &lt;filename&gt; in the directory that the original video file was stored. The generated .mat files, videos, figure files, and original video file will be moved into the folder.</p> <p>Input options are y or n. If y is selected, a folder will be made and files generated during the code will be moved into that folder. If n is selected, the code section is skipped. If something other than y or n is selected, the program will execute as if n was chosen.</p>

### Output

**Table F-3:** Output files for flame front extraction code with associated descriptions.

Output	Description
<filename>_area.mat	MATLAB output file containing all of the variables generated when running the code
<filename>_areavid.mp4	Video showing the original video and reaction front outline without considering inactive points in the reaction front.

<code>&lt;filename&gt;_veloc.fig</code>	Optional file, velocity histogram.
<code>&lt;filename&gt;_area.fig</code>	Figure showing the LFlame/LFrame for the video as a function of time.
<code>&lt;filename&gt;_flametrack.mp4</code>	Video showing the original video and reaction front outline. Red line is the entire reaction front including inactive points, white line is only active points in the reaction front, blue lines are velocity vectors between frames.

### ***F.3. Variable description***

A full list of variables, their data types, and their descriptions is provided below for reference when making modifications to the code or debugging the program. The variables are listed in alphabetical order.

**Table F-4:** Variable list and description for flame front extraction code.

<b>Variable</b>	<b>Description</b>
<code>B</code>	bwboundaries output that has a list of all of the regions in the video based on the runningfilled variable.
<code>binsens</code>	Binary sensitivity variable used in imbinarize.
<code>bitmap</code>	Bitmap output from MatCine.
<code>blackLevel</code>	Default black level of the camera as determined by manufacturer (64).
<code>blackLevelFactor</code>	Factor for increasing the black level in the video for the im_construct variable. User determined.
<code>boundary</code>	Pixel list of the boundaries output by bwboundaries.
<code>bright</code>	Logical array that denotes the bright region in the frame.
<code>brightframe</code>	Brightest frame in the video determined by the light intensity variable calculation.
<code>bw</code>	Initially binarized array prior to padding and hole removal.
<code>closestpt</code>	Cell array showing the closest point between an active point in the reaction front in the current frame and the next frame.
<code>cms_ms</code>	User input variable to use either cm/s or m/s as the x-axis in the local velocity histogram.

<code>crop_x</code>	Cropping window in x, only used when performing fractal analysis.
<code>crop_y</code>	Cropping window in y, only used when performing fractal analysis.
<code>dark</code>	Logical array that denotes the dark region in the frame.
<code>dist</code>	Array calculated when finding the nearest pixel for velocity calculations. Units are in pixels.
<code>edgelen</code>	Calculated length of pixels along the border of the video to remove from the calculated reaction front length.
<code>ext</code>	File extension of selected file
<code>f</code>	Figure handle.
<code>filename</code>	Full filename with the path and filename string.
<code>files</code>	Selected file from GUI.
<code>filled</code>	Binarized image of the active frame with edge holes removed.
<code>filled_hist</code>	A variable that is <code>im_height</code> x <code>im_width</code> that determines where the samples have already burned. Logical datatype.
<code>frac_dim</code>	Calculated fractal dimension for the frame. Each cell in array is for a different frame.
<code>frac_dim_avg</code>	Average fractal dimension of the reaction front over the whole video.
<code>frameno</code>	Annotation handle for figure to denote the frame number being shown.
<code>frame_end</code>	Starting frame for velocity histogram analysis, user input.
<code>frame_start</code>	Final frame for velocity histogram analysis, user input.
<code>framerate</code>	Framerate either determined from setup values in MatCine or user entered.
<code>front_chainlength</code>	Cell variable with perimeter for each measured chain of pixels in the frame.
<code>front1</code>	Cell variable with all of the pixels in the front concatenated into a single cell. Each column is a different frame. Border pixels and NaNs are removed.
<code>fronts_chain</code>	Cell array with the reduced fronts copied. Used only when changing the normalization window.
<code>fronts_og</code>	Copy of the <code>front1</code> variable, <code>og</code> is to denote that they are the originally determined fronts.
<code>fronts_red</code>	Cell variable reduced from <code>fronts_og</code> by removing pixels that were inactive for 3 frames.
<code>gain</code>	Gain for the rendered videos, determined either automatically or by the user.

header	Header output from MatCine
holesize1	Hole size value for removal of random holes from binarized image, empirically chosen.
holesize2	Hole size value for removal of random holes from binarized image, empirically chosen.
i	Counting variable.
im_construct	Copied image array that is manipulated for black level filtering and other calculations.
im_height	Image height of video.
im_width	Image width of video.
j	Counting variable.
k	Counting variable.
light_intensity	Summarized light intensity. Each cell in the array is for a different frame.
long_chain	User input variable to retain only the longest chain of continuous pixels in the reaction front for later calculations.
n	Boxcount output. Only used when performing fractal analysis.
name	Name of selected video.
norm_y	User input variable to normalize by the entire window width or by the range of y values in the reaction front.
num_images	Number of images in the video.
og	Original colored and demosaiced video and increased by the user-defined gain.
path	Path of selected file from GUI.
pathstr	Path in string format.
perim	Variable of calculated perimeter for the frame. Each cell is for a different frame, units are in pixels.
pixel_dist	Calculated pixel distance between two pixels.
pixNumel	Number of pixels in a video.
plot_vectors	Option for users to display the vectors for local velocity in the videos for flame front tracking.
pxSize	Pixel size in $\mu\text{m}/\text{px}$ .
r	Boxcount output. Only used when performing fractal analysis.
raw_image_array	Three-dimensional array of pixels, image height x image width x frame.
rect2	Cropping rectangle. Only used when performing fractal analysis.
runningfilled	Three-dimensional variable of im_height x im_width x frame that contains the filled_hist video as a function of time. Logical datatype.
setup	Setup information from MatCine, contains recording parameters.



tempfront	Temporary front variable that is used to see whether or not a pixel is considered to be part of the active reaction front.
time	Array of timestamps.
timelab	Annotation handle for figure to denote the timestamp of the frame being shown.
turb_ratio	Calculated values for LFlame/LFrame using the normalization size picked by the user. Each cell in the array is for a different frame.
unburned_right	Calculated value for the unburned area region.
v	Video handle.
vbound	Cell datatype with all of the different regions in a frame determined by the bwboundaries and their associated pixels. Each column is a different frame, each row is a different region.
veloc	Structure of velocities in the
velocvert	Vertically concatenated array of all velocities in veloc over the frames between frame_start and frame_end. Used for histogram plotting purposes.
xbotlen	Length along x in bottom border for active frame.
xtoplen	Length along x in top border for active frame.
y	Counting variable.
y_height	Range of y values in a given frame, used when the reaction front length is normalized to something other than the window height.
yllen	Length along y in left border for active frame.
yrllen	Length along y in right border for active frame.
z	Counting variable.

#### ***F.4. Code structure***

The flame front extraction software is broken up into discrete sections as outlined in the code itself and separated by `%%`. A description of each of these sections and their main functions is below, but the program has been heavily commented for further reference.

**Table F-5:** Code structure for flame front extraction code.

<b>Section</b>	<b>Description</b>
Introduction and organization	Clearing entire workspace and adding folders/subfolders for dependencies to the working path.

1. Import of Images	Importing images using MatCine, extracting variables necessary for blacklevel filtering, and pixel size determination.
2. Flame front length and determination	The initial working section of the code that takes the images in from the previous section, binarizes the image, determines the flame front location using the written routines, and then creates a video showing the outline of the flame front. Variables generated in this section are necessary for the next section to make accurate reaction front estimations.
3. Calculating propagation rate for each point	<p>This section complements the previous section by refining the flame front locations. A struct is made to determine whether or not a location in the reaction front was active by checking how long a single pixel is listed in the active front. The reaction front length is then determined using the refined reaction front locations. The determined reaction front length is then normalized by a user-determined length.</p> <p>This section also calculates the local velocity by iterating through each point in the active front, determining the nearest active point in the reaction front in the next frame, and then dividing by the time change.</p> <p>A video is made where the active reaction front points are shown in white and the inactive reaction front points are shown in red. Blue lines are the velocity vectors of the points in the active reaction front between frames.</p>
4. Calculating fractal dimension in each frame using boxcount (beta)	Optional code section to determine the fractal dimension using the boxcount algorithm.
5. Figure creation showing the LFlame/LFrame for the video	Code section that creates a figure of LFlame/LFrame vs. time for the video length.
6. Figure creation showing the local velocity distribution for the video	Optional code section to create a histogram of the local velocities that was determined in the previous sections. Users must know the starting and ending frames that they would like to include in the histogram, and it is recommended that only frames where the whole reaction front length is present in the window and the velocity vectors do not lead to the edge of the video.

7. Clean Up Directories	An option for the user to clean up directories in their working directory by creating a new dated folder and moving all newly generated files into that folder along with the original video that was used for analysis.
-------------------------	--

### ***F.5. Summary***

This user manual and code description includes the input, output, and structure of the flame front extraction code developed for high-speed microscopy research of reacting materials, but the general concepts here for image binarization and region recognition could be extended to other disciplines or studies. If sections of this code are used in other programs or should be cited as part of future research, please reference either this dissertation or D.J. Kline, M.C. Rehwoldt, H. Wang, N.E. Eckman, M.R. Zachariah, Why does adding a poor thermal conductor increase propagation rate in solid propellants?, Appl. Phys. Lett. 115 (2019) 114101. doi:10.1063/1.5113612.

### ***F.6. Version history***

A simple version history is presented below for reference and v10 is provided in the following section.

**Table F-6:** Version history for flame front extraction code.

<b>Version</b>	<b>Date</b>	<b>Description</b>
1	8/23/18	Initial version, binarizes image and shows video.
2	8/29/18	Refined v1.
3	8/30/18	Added new section to determine local velocities.
4	8/31/18	Refined section to determine active points in reaction front based on 3-frame change. Used this to refine reaction front length and local velocity calculations.
5	3/22/19	Bug fixes.
6	3/22/19	Bug fixes.
7	8/30/19	Bug fixes.
8	10/3/19	Introduced option to calculate fractal dimension in main code.

9	11/18/19	Bug fixes, code refining.
10	2/1/20	Add option to choose to measure only the longest frame and use a custom normalization height.

### ***F.7. MATLAB code***

```

%% Time-Resolved Flame Front Measurements for Microscopy
% Dylan J. Kline
% Chemical Engineering - Zachariah Research Group
% University of Maryland College Park

%% Introduction and Organization
% This program was designed to spatiotemporally resolve the flame
front of
% turbulent reaction fronts in energetic films prepared via additive
% manufacturing methods (3D printing, E-spin, E-spray, etc.). Videos
% recorded are collected with a Phantom Vision Research Camera
(notably
% the VEO710L or Miro M110).
%
% Dependencies:
% MatCine (v1.0, Dylan Kline) [found in External Functions Folder].
% Image Analysis Toolbox (Matlab)
% Boxcount (for fractal analysis) [found in External Functions
Folder].

% Samples and experiments which inspired the creation of this
program have
% come from Rohit Jacob, Haiyang Wang, and Miles Rehwoldt. Their
input at
% every step of the way in designing the experiment to preparing
samples
% is greatly appreciated.

clc
clear
close all
tic

% Adds all external functions to working path.
addpath(genpath(fileparts(which(mfilename))))

%% 1. Import of Images
% Images are imported using a house-built function (MatCine) which
was
% previously written in Python. As of Aug. 2018, videos recorded on
PCC
% v2.8 and above are compatible. If the video is not compatible,
resave
% the video in a newer version of PCC.

% Pick .cine or .mat file using ui - multiple files can be selected
[files, path]=uigetfile({'*.cine;*.mat'}, 'Pick Cine File');
filename=fullfile(path, char(files));
[pathstr,name,ext]=fileparts(filename);

% get the original image from before focusing
% pre_img = imread(fullfile(path, char(uigetfile('*.bmp'))));

```

```

% Extraction of images using pycine (before) or reruns
if ext(end-3:end)==' .mat'

    load(filename,'raw_image_array')           % should import
the data as a matlab file
    %raw_image_array=uint16(raw_image_array);    % converts to
uint16 for faster processing

else

    [header, bitmap, setup, raw_image_array]=MatCine(filename);

end

% Details about video for later calculations
im_height=size(raw_image_array,1);           % extracts image width
im_width=size(raw_image_array,2);            % extracts image
height
pixNumel=im_width*im_height;                 % gets pixels per
image
num_images=size(raw_image_array,3);           % calculates number of
images in file.
im_construct=raw_image_array;                 % makes an
im_construct later

% Correction for black level
% This feature has been introduced in the case that someone has a
video
% that is darker than usual or finds that a default value does not
match
% the visually-perceived outline of the flame. Typically, this would
% happen when there is light scattering around the edges of the
slide, but
% there are a variety of reasons someone would need to change the
black
% level.
blackLevel=64;
if input('\nBlack level factor\nDefault (1) or other (2)? : ')==1
    blackLevelFactor = 1.5; %
Default value was chosen empirically from experiments with the color
camera for pyrometry
else
    blackLevelFactor = input('Black level factor (1.5 default): ');
end
im_construct(im_construct(:,:, :)<blackLevel*blackLevelFactor)=0; %
A non-default option has been used when the black level appears to be
failing the point of the code.

% Time Setup
if ext(end-3:end)==' .cine'
    framerate=setup.Framerate16;
else
    framerate=input('Framerate (pps): ');
end

if framerate>0
    time=(0:num_images-1)./framerate;
else
    time=(0:num_images-1);
end

```

```

% Determining the pixel distance for the video recorded
if input('\nSize calibration\nlum/px (1) or 1.7um/px (2):')==1
    pxSize = 1;          % Old microscope assembly with 1um/px
    resolution
else
    pxSize = 1.7;        % K2 microscope assembly with 1.7um/px
    resolution
end

%% 2. Flame front and length determination

% making a normalized light intensity calculation
for i=1:num_images
    % Integrated amount of light hitting the sensor
    light_intensity(i)=sum(sum(sum(im_construct(:, :, i))));
end
light_intensity=light_intensity/max(light_intensity);

% Gain adjustment to make videos brighter, user defined.
% This feature was added for the cases where the videos were
naturally
% bright and that a "max gain" would artificially change the
perceived
% edge in the video (i.e. calculated front is far away from the
measured
% front because of pixels getting brighter further away by light
% scattering). The options are to have the gain autocalculated so
the
% brightest pixel in all of the videos becomes "saturated" and all
of the
% other pixels scale with the ratio from the brightest pixel to a
% saturated value. The "max gain" option just makes the gain a
straight
% value of 10 which is the maximum that PCC allows in the program as
of
% 11/19/19.
[~,brightframe]=max(light_intensity);
if input('\nVideo gain\nGain auto (1) or gain max (2)?')==1
    gain=4096/max(max(raw_image_array(:, :, brightframe))); %
    auto gain routine that is used in the pyrometry code. Makes brightest
    pixel white.
else
    gain=10; %
    makes the gain automatically a value of 10. Best for really dark
    videos.
end

% Testing subtracting the first frame since these films have some
serious hole problems
im_construct = bsxfun(@minus, im_construct, im_construct(:, :, 1));

% Binary Thresholding value
binsens = 1;
holesize1 = 1000;
holesize2 = 5000;

% Initialization of the "history" of the flame front
bw = zeros(im_height, im_width);
%ed = zeros(im_height, im_width);

% Making a video
v=VideoWriter(sprintf('%s_areavid.mp4', horzcat(path, name)), 'MPEG-4');

```

```

v.FrameRate=10;
v.Quality=100;
open(v)
figure('Color',[1 1 1]);

% initializing a history
filled_hist = zeros(im_height,im_width);

for k=1:num_images

    og = double(demosaic(gain*im_construct(:, :, k), 'gbrg'))/4096;
% demosaics the original image from the camera and corrects it for
gain and the bit depth
    %og = imfilter(og,2);
    bw = imbinarize(rgb2gray(og),
'adaptive','sensitivity',binsens,'ForegroundPolarity','bright');
    filled = padarray(bw,[1 1],1,'both');
% padding the image helps with hole removal on the edges of the video
    filled = bwareaopen(filled,holesize1);
% fill holes in the image that are less than a desired size, mostly
takes care of large edge problems
    filled = filled(2:end-1,2:end-1);
% remove the artificial pad from the image

    filled_hist = filled + filled_hist;
% creates a summed model of the history of the flame front to make
effectively one large front
    filled_hist = imfill(filled_hist,'holes');
% fills holes in the image since that was a problem
    runningfilled(:, :, k)=filled_hist;
% keeps a running filled history of the images for later image
processing
    filled_hist = bwareaopen(padarray(filled_hist,[1
1],1,'both'),holesize2); % fills holes that are on the edges by first
padding the array and then filling holes, this took some finessing
    filled_hist = filled_hist(2:end-1,2:end-1);
% recropping the image back to its original size after padding the
array to fill the holes

    % still tryin to remember why i did this
    dark=regionprops(~filled_hist, 'Area','PixelList','BoundingBox');
% create region profiles for the dark spots that are unburned (either
before or after front)
    unburned_right = max(vertcats(dark.Area));
% logically, the largest unburned (or dark) area would be in the
direction of propagation, so extract area for later processing

    bright=regionprops(filled_hist, 'Area','PixelList','BoundingBox');

% Making a figure
    image(double(demosaic(gain*raw_image_array(:, :, k), 'gbrg'))/4096);
% show the original color image

    hold on

        if size(bright,1)>=1
calculate the perimeter

            % This small subroutine removes the length of the frame
edge

```

```

        % from the value estimated as part of the perimeter since
it
        % really isnt a part of the reaction zone.
        yllen=sum(filled_hist(:,1)); % find
length vertically along left edge
        yrllen=sum(filled_hist(:,end)); % find
length vertically along the right edge
        xtoplen=sum(filled_hist(1,:)); % find top
length horizontally along edge
        xbotlen=sum(filled_hist(end,:)); % find
bottom length horizontally along edge
        edgelen = yllen + xtoplen + xbotlen+yrllen; % total
area accountable for edge

        % adopted from online
https://www.mathworks.com/matlabcentral/answers/301666-how-to-find-length-or-perimeter-of-a-irregular-boundary
        % dropped using the perimeter function because there is
evidence that it does some sort of interpolation between pixels
        % https://www.mathworks.com/matlabcentral/answers/357994-perimeter-used-by-regionprops
        % effectively this does a pythagorean length calculation
over
        % the whold boundary and sums those lengths for each
boundary
        B = bwboundaries(runningfilled(:,:,k));
% bwboundaries does analysis on the image to separate into boundaries
(faster than regionprops)
        perim(k)=0;
% reinitialize the perimeter as part of the matrix

        for j=1:length(B)
% go through bwboundaries individually
            boundary = B{j};
            plot(boundary(:,2), boundary(:,1), 'w', 'LineWidth',1);
% show boundary on image
            vbound{j,k} = boundary;

        end
    end

    % Annotations to make labels for time, frame number, and the
effective area of the flame in microns
    if k==1
        frameno=annotation('textbox', [0.225, 0.1, 0, 0], 'string',
sprintf('frame
%d',k), 'FitBoxToText', 'on', 'FontSize',12, 'BackgroundColor', 'w');
        timelab=annotation('textbox', [0.34, 0.1, 0, 0], 'string',
sprintf('t=%4.3fms',time(k)*1000), 'FitBoxToText', 'on', 'FontSize',12, '
BackgroundColor', 'w');
    else
        timelab.String=sprintf('t=%4.3fms',time(k)*1000);
        frameno.String=sprintf('frame %d',k);
    end

    % Annotations to make labels for time, frame number, and the
effective area of the flame in microns
    if k==1
        frameno=annotation('textbox', [0.225, 0.1, 0, 0], 'string',
sprintf('frame
%d',k), 'FitBoxToText', 'on', 'FontSize',12, 'BackgroundColor', 'w');

```



```

        timelab=annotation('textbox', [0.34, 0.1, 0, 0], 'string',
sprintf('t=%4.3fms',time(k)*1000),'FitBoxToText','on','FontSize',12,'
BackgroundColor','w');
    else
        timelab.String=sprintf('t=%4.3fms',time(k)*1000);
        frameno.String=sprintf('frame %d',k);
    end

    daspect(gca,[1 1 1])
    hold off

    % Title of the figure
    title(sprintf('Flame Front
Tracking\n%s',name),'interpreter','None')

    %Remove tick labels
    set(gca,'xtick',[]);
    set(gca,'ytick',[]);

    % Spielberg
    f=getframe(figure(1));           % get video frame location
    writeVideo(v,f)                  % append image

    clearvars front

    pause(0.25)

end
close all
close(v)

%% 3. Calculating propagation rate for each point
clearvars ve2 veloc fronts_chain
%fronts_red{1}=[NaN,NaN];
perim=zeros(1000,1000);

% Making a cell of the fronts because I was going crazy
for k=1:num_images
    front1 = [NaN,NaN];
    i=1;
    while i<=size(vbound,1) && isempty(vbound{i,k})==0 % iterate
through the cell structure for flame fronts
        front1 = vertcat(front1,vbound{i,k}); % extract the
first flame front (frame i), vertically concatenate to 1D array
        i=i+1; % increment
    end
    front1(isnan(front1(:,2)),:)=[];
    front1(front1(:,1)==1,:)=[]; % removing
edges in height at 1
    front1(front1(:,1)==im_height,:)=[]; % removing
edges in height at max height
    front1(front1(:,2)==1,:)=[]; % removing
edges in width at 1
    front1(front1(:,2)==im_width,:)=[]; % removing
edges in width at max width
    fronts_og{k}=front1;
    clearvars front1
end

front1 = [NaN,NaN];

% Determines if the user wants to only retain the longest chain

```

```

long_chain = input('Retain only longest chain? (y/n): ','s');
norm_y = input('Use entire frame as normalization length? (y/n): ','s');
plot_vectors = input('Display vectors for local velocity on plot? (y/n): ','s');

% Video recorded
v=VideoWriter(sprintf('%s_flametrack.mp4',horzcat(path,name)),'MPEG-4');
v.FrameRate=10;
v.Quality=100;
open(v)
figure('Color',[1 1 1]);

for k=1:num_images-1

    % This section checks to make sure that the point is moving
    rather than
    % not moving
    m=1;
    % initializing variable for point-by-point iteration
    tempfront(1,:)=NaN,NaN];
    % temporary variable to keep track of the front
    if isempty(fronts_og{k})==0
    % checks to see if the front is empty
        for m=1:size(fronts_og{k},1)
    % iterates point-by-point in each of the fronts
            if ismember(fronts_og{k}(m,:),fronts_og{k-1},'rows')==1
    % see if the point in the current front was part of the front in the
            last frame
                if k>=3
    % if the frame number is larger than 3 (since we check the last two
                frames)
                    if ismember(fronts_og{k}(m,:),fronts_og{k-2},'rows')==1 &&... % see if the point in the front is present
                    in the front two frames ago
                        ismember(fronts_og{k}(m,:),fronts_og{k+1},'rows')==1 %
                        see if the point in the front is present in the next frame
                        continue
                    end
                end
            end
        else
            tempfront=[tempfront; fronts_og{k}(m,:)];
    % if the point was 'active', add it to the list of active fronts
        end
    end
end
tempfront(isnan(tempfront(:,2))==1,:)=[];
% remove all NaN's from the temp front
fronts_red{k}=tempfront;
% temporary front saved as a new variable called fronts_red (for
reduced-length fronts)

clearvars tempfront

% removing NaNs
% fronts_red{k}(isnan(fronts_red{k}(:,2))==1)==[];

% calculating reduced perimeter with a filter for distances
greater
% than 2 (indicating a break in the segments of the front)

```

```

    % creates a chain variable that identifies the different lengths
    that
    % are present in the front
    z = 1;
    y = 1;
    if k>1
        for h = 1:length(fronts_red{k})-1
            % calculate perimeter pixel by pixel
            if size(fronts_red{k},1)>=2
                % determines if there is actually a front there
                pixel_dist = ((fronts_red{k}(h,1) -
                    fronts_red{k}(h+1,1)).^2 + (fronts_red{k}(h,2) -
                    fronts_red{k}(h+1,2)).^2).^0.5; % determine the distance between two
                pixels in the front
                if pixel_dist>2
                    % a pixel distance greater than 2 would suggest some break in the
                    front, skip
                    front_chainlength{z,k} = perim(z,k);
                    % Retain a chain length for each chain calculated
                    z = z+1;
                    % start new chain
                    y = 1;
                    % restart counter for chain
                    continue
                    % skip counting
                else
                    fronts_chain{z,k}(y,:) = fronts_red{k}(h,:);
                    perim(z,k) = perim(z,k) + pixel_dist;
                    % add distance to the perimeter tally
                    y = y+1;
                end
            end
        end
        else
            perim(k,z)=NaN;
            % If there wasn't a front to really measure (like a black frame)
        end

        % Handling whether or not only the longest chain should be
        retained
        if long_chain == 'y' % if the user
            decided to keep the longest chain
            [perim(1,k),z] = nanmax(perim(:,k),[],1); % identify the
            longest chain and the index of the longest chain
            perim(2:end,k) = NaN; % make everything
        else NaN
            if exist('fronts_chain')==1 % if the
                fronts_chain exists
                if k<=size(fronts_chain,2)
                    fronts_red{k} = fronts_chain{z,k}; % retain only the
                    longest chain and use it in the reduced fronts
                end
            end
            else % if not, just
                sum the chain lengths
                perim(1,k) = sum(perim(:,k),1);
            end

            % determines the length of the perimeter relative to the image
            length
            if norm_y ~= 'n'
                turb_ratio(k) = perim(1,k)/im_height; % if the user
                decided to use the frame height as the normalization length
            end
        end
    end
end

```

```

else
    if isempty(fronts_red{k})==0 % else, if the
user decided to use a changing window length because of some other
reason
        y_height(k) = range(fronts_red{k}(:,1)); % determine the
vertical range of the flame front
        turb_ratio(k) = perim(1,k)/y_height(k); % normalize the
length based on that vertical range
    end
end

% makes a graph
image(double(demosaic(gain*raw_image_array(:, :, k), 'gbgr'))/4096);
% show the color image
hold on
% for figure creation
scatter(fronts_og{k}(:,2), fronts_og{k}(:,1),1,'r');
% plot the original whole length front in red
%scatter(fronts_og{k+1}(:,2), fronts_og{k+1}(:,1),1,'g');
% shows the next front in the video for comparison
if isempty(front1)==0
    scatter(fronts_red{k}(:,2), fronts_red{k}(:,1),1,'w');
% overlay the active front lengths in white
end

% loop to find closest point and calculate velocity
if isempty(fronts_red{k})==0 && isempty(fronts_og{k+1})==0
% makes sure there is actually a front to measure
    for m=1:size(fronts_red{k},1)
% iterates point-by-point to make velocity estimate
        dist = bsxfun(@hypot, fronts_og{k+1}(:,1)-
fronts_red{k}(m,1), fronts_og{k+1}(:,2)-fronts_red{k}(m,2)); %
perform an operation to estimate distance from one pixel in one front
to each pixel in the next front
        closestpt{k}(m,:) =
fronts_og{k+1}(find(dist==min(dist),1,'first'),:);
% find the closest point
        veloc{k}(m,1) = min(dist)*1E-4*pxSize/(time(k+1)-time(k));
% estimate the velocity using the distance and dividing by time

        if plot_vectors == 'y'
            plot([fronts_red{k}(m,2),
closestpt{k}(m,2)], [fronts_red{k}(m,1), closestpt{k}(m,1)], 'c');
        end
    end

    end
    veloc{1,k}(isnan(veloc{1,k})==1)=[];
% collect the velocities in a cell array and removes NaN's
end

% Figure things
daspect(gca,[1 1 1])
hold off

% Annotations to make labels for time, frame number, and the
effective area of the flame in microns
if k==1
    frameno=annotation('textbox', [0.225, 0.1, 0, 0], 'string',
sprintf('frame
%d',k), 'FitBoxToText', 'on', 'FontSize',12, 'BackgroundColor', 'w');
    timelab=annotation('textbox', [0.34, 0.1, 0, 0], 'string',
sprintf('t=%4.3fms',time(k)*1000), 'FitBoxToText', 'on', 'FontSize',12, '
BackgroundColor', 'w');

```

```

        arealab=annotation('textbox', [0.47, 0.1, 0, 0], 'string',
sprintf('Area Ratio:
%1.2f',perim(k)/im_height), 'FitBoxToText', 'on', 'FontSize',12, 'Backgro
undColor','w');
    else
        timelab.String=sprintf('t=%4.3fms',time(k)*1000);
        frameno.String=sprintf('frame %d',k);
        arealab.String=sprintf('Area Ratio:
%1.2f',perim(k)/im_height);
    end

    %Remove tick labels
    set(gca,'xtick',[]);
    set(gca,'ytick',[]);

    % Title of the figure
    title(sprintf('Flame Front
Tracking\n%s',name), 'interpreter','None')

    % Spielberg
    f=getframe(figure(1));           % get video frame location
    writeVideo(v,f)                 % append image

    clearvars front

end
close all
close(v)

%% 4. Calculating fractal dimension in each frame using boxcount
% Well, this originally started as a good idea but slowly devolved
into
% something else. We weren't really sure if the reaction front was
truly a
% fractal or some other just very corrugated structure. Previous
attempts
% to calculate this on older videos suggested a fractal dimension
~1.2, so
% if someone in the future finds something interesting that would be
% awesome. If not, it will just be another cool but rather useless
% feature.

if input('Perform fractal analysis? (y/n): ','s')== 'y'

    % Initially setting this up, it was found that cropping the video
to
    % avoid any weirdness at the edges was the best policy.

    [~,rect2]=imcrop(image(double(demosaic(raw_image_array(:, :,brightfram
e)*gain, 'gbgr'))/4096));
    crop_x = round([rect2(1) rect2(1)+rect2(3)]);
    crop_y = round([rect2(2) rect2(2)+rect2(4)]);

    for k=1:size(fronts_red,2)
        if isempty(fronts_og{k})==0
            tempfront=fronts_red{k};           % creates a
temporary front array that uses only cropped values
            tempfront(tempfront(:,1)<crop_x(1),:)=[]; % cropping in
x
            tempfront(tempfront(:,1)>crop_x(2),:)=[]; % cropping in
x

```

```

        tempfront(tempfront(:,2)<crop_y(1),:)=[];    % cropping in
y
        tempfront(tempfront(:,2)>crop_y(2),:)=[];    % cropping in
y
        z = zeros([im_height im_width]);            % creating a
0 matrix with desired dimensions
        for i=1:size(tempfront,1)
            z(tempfront(i,1),tempfront(i,2))=1;      % iterate
through front and add tempfront into image as white line
        end
        [n,r]=boxcount(z);                            %
perform boxcount operation to estimate fractal dimension
        frac_dim(:,k) = (-gradient(log(n))./gradient(log(r)))'; %
plot of fractal dimension
        frac_dim_avg(k) = nanmean(frac_dim(:,k));      %
plot of average fractal dimension
        clearvars n
    else
        continue;
    end
end

    % Plots the fractal dimension for all of the frames in the video
on the
    % same graph to see if there is any natural resting point.
Generally
    % there is.
    fig=figure('Color',[1 1 1]);
    hold on
    for k=1:size(fronts_red,2)
        plot(r,frac_dim(:,k),'LineWidth',2)
    end
    xlabel('r, box size');
    ylabel('-d(ln N)/d(ln r), local dimension');
    set(gca,'FontSize',16,'XScale','log')
    hold off
end

%% 5. Figure creation showing the LFlame/LFrame for the video
fig=figure('Color',[1 1 1]);
set(fig, 'defaultAxesColorOrder',[0 0 0]; [1 0 0])
hold on
plot(1000*time(1:length(turb_ratio)),turb_ratio,'LineWidth',2)
xlabel('time (ms)')
ylabel('L_{flame} / L_{frame}')
xlim([1000*min(time) 1000*max(time)])
title(sprintf('Flame Front Area\n%s',name),'interpreter','None')
grid('on')
ax=gca;
ax.FontName='Times New Roman';
ax.FontSize=16;
ax.FontWeight='bold';
hold off

savefig(sprintf('%s_area.fig',horzcat(path,name)))

%% 6. Figure creation showing the local velocity distribution for the
video
% This section creates a histogram with the locally calculated
velocities
% based on a closest-point distance analysis. It should be noted
that this

```

```

% section should only be run if the user knows which frames have
good
% data... For example, as the film propagation approaches the edges
of the
% slides, you might find some unrealistic values for velocity since
it is
% just estimating the edge to be the end. Random jumps in the film
are also
% unrealistic. It is suggested that users run the code once, make
sure the
% data looks like what they want it to, then rerun it with this
section
% and note the frames that are good.

if input('Create velocity histogram? (y/n): ','s')== 'y'

    velocvert = [NaN]; %
    initializing the vertically-concatenated velocity array
    frame_start = input('Start frame for velocity analysis: '); %
    user defined starting image for analysis
    frame_end = input('End frame for velocity analysis: '); % user
    defined ending image for analysis
    for k=frame_start:frame_end %
    loop to vertically concatenate the velocities
        velocvert = [velocvert; veloc{1,k}]; %
    extract the first flame front (frame i), vertically concatenate to 1D
    array
    end

    velocvert(velocvert==0)=[]; %
    removes zero velocities since that isnt real and would distort the
    average

    cms_ms = input('X-axis in cm/s (1) or m/s (2)? : ');

    % Figure creation and saving of the histogram.
    fig=figure('Color',[1 1 1]);
    set(fig, 'defaultAxesColorOrder',[0 0 0]; [1 0 0])
    hold on

    if cms_ms == 0
        h=histfit(velocvert,100,'lognormal');
        xlabel('Velocity (cm/s)')
    else
        h=histfit(velocvert/100,100,'lognormal');
        xlabel('Velocity (m/s)')
    end

    set(h(1), 'facecolor', 'w');
    set(h(2), 'color', 'b')

    %vline(nanmean(ve2), 'r', sprintf('Average=%1.2fcm/s', nanmean(ve2)));
    ylabel('Count')
    title(sprintf('Velocity Distribution
\n%s', name), 'interpreter', 'None')
    grid('on')
    ax=gca;
    ax.FontName='Times New Roman';
    ax.FontSize=16;
    ax.FontWeight='bold';
    hold off

```

```

        savefig(sprintf('%s_veloc.fig',horzcat(path,name)))

end

%% 7. Clean Up Directories
save(char(sprintf('%s_area.mat',horzcat(path,name))), 'binsens','front
s Og','fronts_red','turb_ratio','veloc','vbound')
cleanup = input('Would you like to clean up directories? (y/n):
','s');
if cleanup=='y'
    basename = filename(1:end-size(ext,2));
    foldername=fullfile(path, sprintf('%s
%s',string(datetime('now','Format','yyyyMMdd')), string(name)));
    mkdir(foldername);

    movefile(sprintf('%s_area.mat',basename),foldername);    % moves
mat file

    if isfile(sprintf('%s_areavid.mp4',horzcat(path,name)))
        movefile(sprintf('%s_areavid.mp4',basename),foldername);    %
moves mat file
    end

    if isfile(sprintf('%s_veloc.fig',horzcat(path,name)))
        movefile(sprintf('%s_veloc.fig',basename),foldername);    %
moves fig file
    end

    if isfile(sprintf('%s_area.fig',horzcat(path,name)))
        movefile(sprintf('%s_area.fig',basename),foldername);    %
moves fig file
    end

    if isfile(sprintf('%s_flametrack.mp4',horzcat(path,name)))
        movefile(sprintf('%s_flametrack.mp4',basename),foldername);
% moves video file
    end

    if exist(sprintf('%s.cine',basename),'file')==2
        movefile(sprintf('%s.cine',basename),foldername);    % moves
cine file
    end
end

```



## Appendix G: T-Jump Nano – A multi-signal pulse generator for precision triggering and heating

### ***G.1. Introduction***

A cornerstone instrument in the Zachariah Group is what is known as the Temperature Jump (T-Jump) Time of Flight Mass Spectrometer (TOFMS). The main feature of this instrument is that it couples and synchronizes mass spectrometry with the rapid heating of a fast response platinum thermocouple coated with a small amount of sample. Currently, a single synchronizing function generator acts as both a single and continuous triggering mechanism. In normal operation, a 1KHz-10KHz TTL square wave with a width of roughly 5s triggers an oscilloscope, an electron gun pulser source, and the heating of a platinum/Nichrome wire. Data collection occurs over 10ms where 10-100 total spectra are collected. In order to achieve this type of data collection, both the oscilloscope and electron gun pulser source must see a continuous trigger signal corresponding to each spectra/ionization time period. However, heating of the wire over a period of 1-100ms requires only a single heating pulse with a variable width and amplitude. As a result, our current setup must utilize a separate heating pulser box powered by a variable power supply which takes the continuous trigger input and outputs a single heating pulse with a calibrated width and amplitude.

### ***G.2. Project purpose***

The purpose of this project is to modernize the current setup while physically consolidating all the major control components, increasing versatility of control, and

enhancing ease of availability and mobility for this type of system to be utilized for other mass spectrometer types and heating schemes.

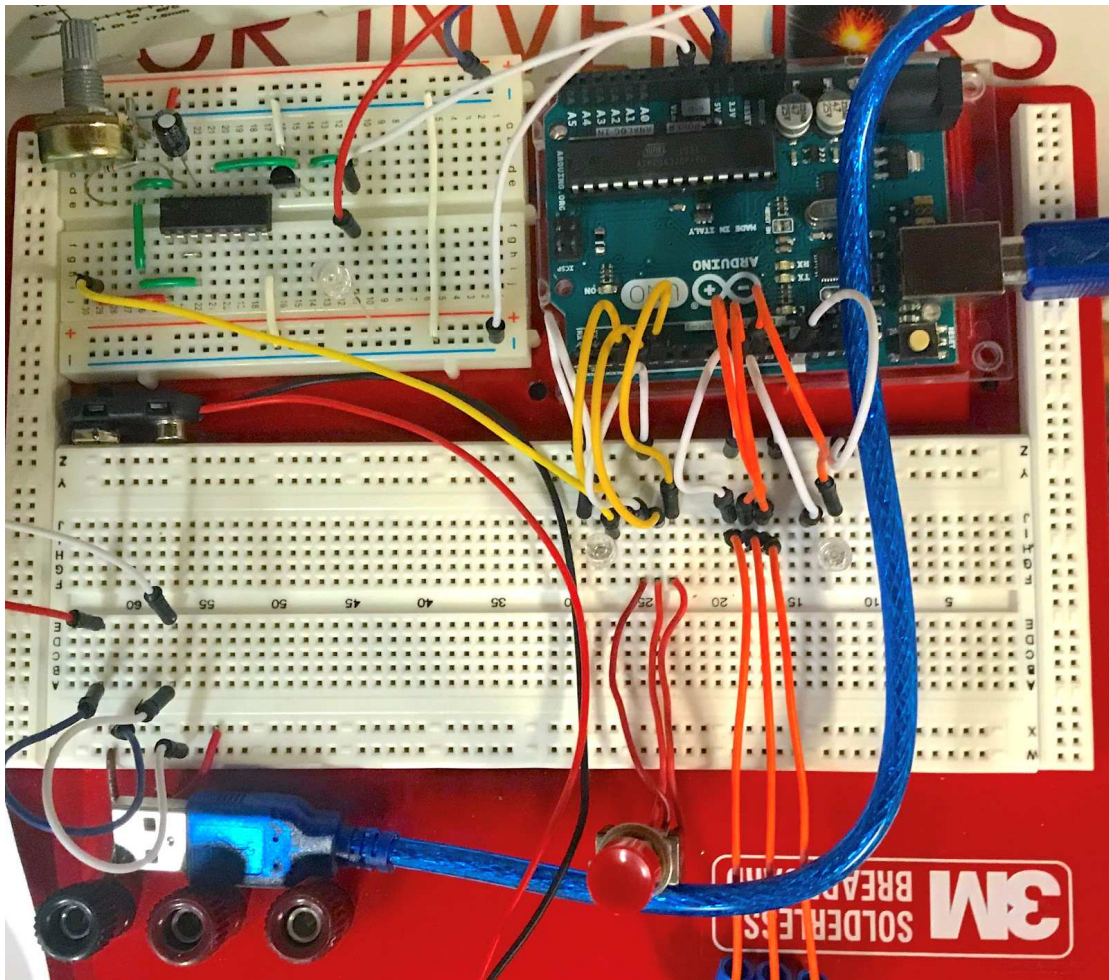
While there isn't anything necessarily wrong with the current setup, most of the components are bulky and lack portability. The system requires a function generator, heating pulser box, oscilloscope, and power supply. All of these components in total require considerable desk space and are also interchangeably used for different experiments in the lab. The current heating scheme is also fairly one dimensional by only allowing a single heating pulse. Currently, there is no way of being able to take the continuous trigger input and output a continuous square wave with a variable width and amplitude for heating purposes. This capability alone would broaden the studies we could undertake with the T-Jump/TOFMS system.

### ***G.3. T-Jump Nano***

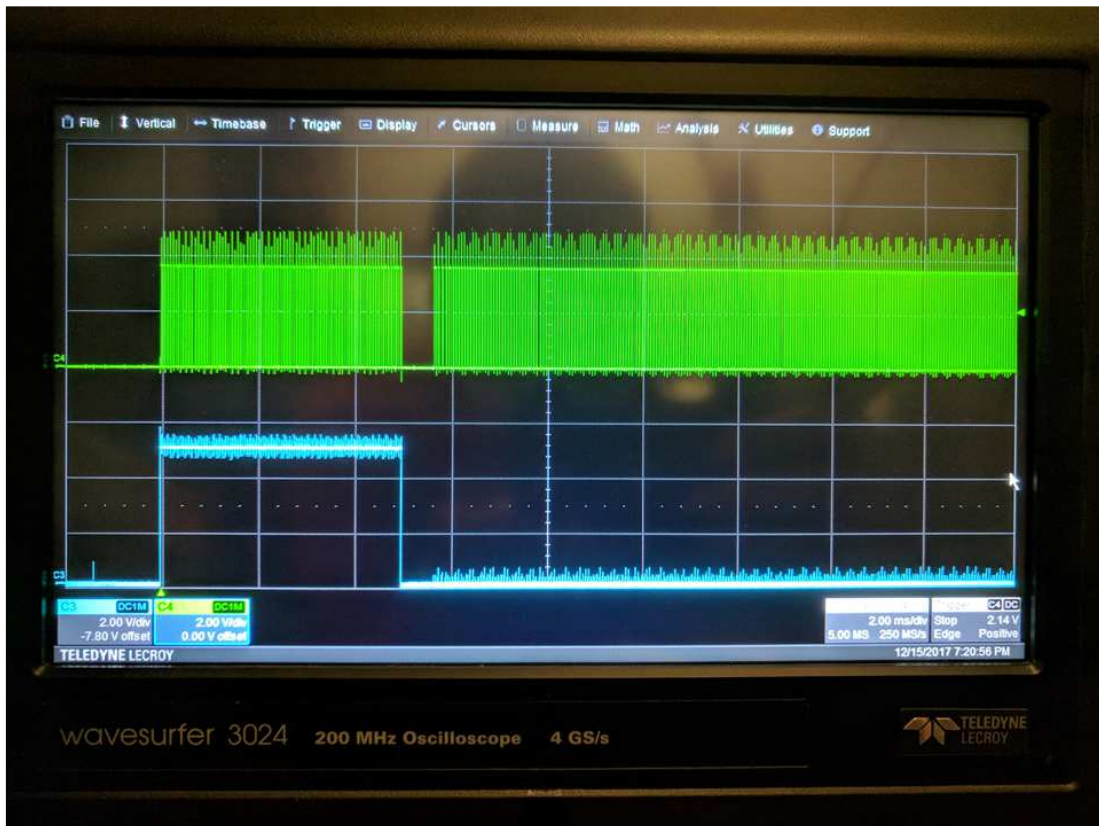
#### ***Initial design and breadboard prototype***

The project design relies on the versatile capabilities of a programmable Arduino microcontroller in place of a function generator. This switch instantly downsizes the current function generator while enhancing control capabilities with a modernized and simplified interface. For this specific design, the Arduino program and circuit design has three BNC output types: a programmable continuous waveform, a programmable single shot, and a modulated heating waveform which could be either single shot or continuous. Switches are implemented complementary to the programmed Arduino action within the circuit scheme in order for all waveforms to be output simultaneously. This allows for the capability of triggering an oscilloscope or

any other electrical component while capturing the data which may come as a result of the single shot output.



**Figure G-1:** Initial solderless breadboard prototype of the T-Jump nano. The breadboard in the top left includes a monostable multivibrator and solid state relay that would actuate a high-voltage pulse to resistively heat a wire. This multivibrator would be modulated with a signal sent by the Arduino Uno (top right) which is controlled with triggers wired in on the breadboard below it.



**Figure G-2:** Arduino output waveforms: repeated (green) and single shot (blue).

Initial experiments to test the functionality of the Arduino Uno and its execution of the trigger's code produced two simultaneous pulses like those in Figure G-2. As can be seen, the code above properly executes a 5ms, single shot pulse (blue) that would be used to trigger the resistive heating of a wire while the other segment of the circuit board continuously produces 20 $\mu$ s pulses (green) until collection by the oscilloscope stops or the trigger is deactivated. When examining the pulses closely, there appeared to be a response resembling those seen in proportional-integral (PI) or proportional-integral-derivative (PID) control schemes. Zoomed images of the initial setpoint change (from LOW to HIGH) was reflected by the output of the device through a sharp step change accompanied by oscillations as the device attempts to control the output voltage internally. When the single shot pulse ends, there appears to be a gap of ~150-

200 $\mu$ s that can be readily explained by the code shown in the figure below. In the section of the code executed from lines 46-64, the code turns the PWM pin 3 to low, which has been benchmarked as lasting 6-10 $\mu$ s, and then the code prints a serial output to indicate the amount of time the single shot pulse was on. This series of actions accounts for the remaining time. In operational conditions, the prototype code shown would eliminate the serial printing since it could be measured much more easily on the oscilloscope directly and the `digitalWrite()` command could be substituted for direct port calls that are benchmarked at  $\sim$ 0.4 $\mu$ s. It is important to note, however, that the typical application of this device calls for 100 $\mu$ s pulses and thus the significance of the gap in the collection by the oscilloscope would be 1-2 samples at maximum.

```

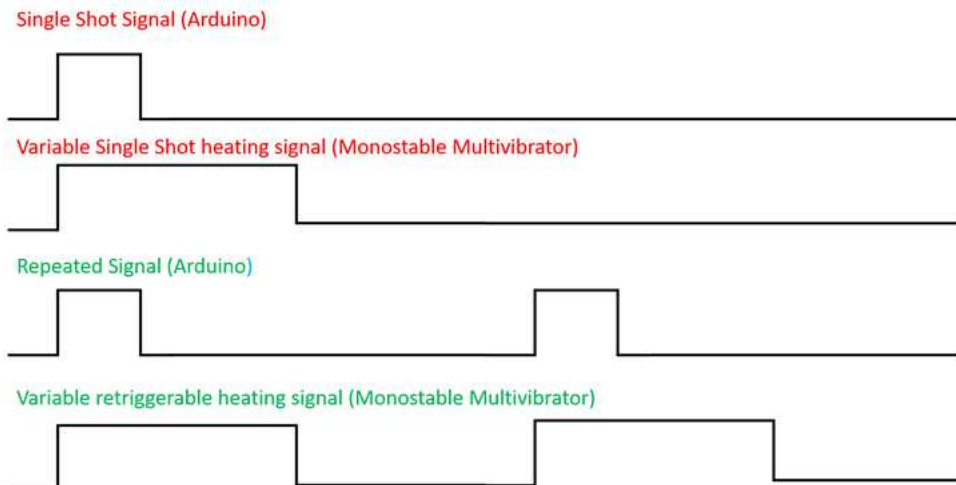
1 void setup() {
2   pinMode(5, OUTPUT); // input to push button for single shot
3   pinMode(4, INPUT);  // output to push button for single shot
4   pinMode(3, OUTPUT); // signal sent to single shot segment of circuit
5   pinMode(11, OUTPUT); // signal sent to repeating pulses
6   pinMode(10, OUTPUT); // input to switch for repeating pulses
7   pinMode(9, INPUT);  // output to switch for repeating pulses
8   Serial.begin(250000); //
9 }
10
11 void loop() {
12
13   static int heat_trig = 0; // pre (0), during (1), or post (2) trigger
14   static int osc = 0;      // determines whether or not to send trigger to oscilloscope
15
16   unsigned long heat_time = 5000; // total time for heating in microseconds
17   static unsigned long heat_start; // stored value for heating start time
18   unsigned long heat_end;         // stored value for heating finish time
19   float pulse;                   // value calculated to determine pulse length
20   static unsigned long current;   // stored value for current time in loop
21
22   digitalWrite(5, HIGH); // sets input of trigger for heating to high
23   digitalWrite(10, HIGH); // sets input of oscilloscope to high
24
25   // if the heating trigger has been pushed and the trigger was reset
26   if (heat_trig == 0 && digitalRead(4) == HIGH) {
27     osc = 1; // turn on oscilloscope triggering signal
28     heat_trig = 1; // indicate internally that the heating trigger was pushed
29     heat_start = micros(); // record the time right before heating in microseconds
30     digitalWrite(3, HIGH); // start heating
31   }
32
33   // if the trigger has been pushed before and the oscilloscope switch is open
34   if (osc == 1 && digitalRead(9) == HIGH) {
35     digitalWrite(11, HIGH); // send pulse to oscilloscope
36     delayMicroseconds(20); // delay execution for 20us
37     digitalWrite(11, LOW); // stop pulse to oscilloscope
38   }
39
40   // if the trigger was pushed, continually record the time
41   if (heat_trig == 1) {
42     current = micros();
43   }
44
45   // if the time the wire has been heating is equal to or greater than the desired time
46   if ((current - heat_start) >= heat_time && heat_trig == 1) {
47     digitalWrite(3, LOW); // stop heating
48     heat_end = current; // record time at which the wire stopped heating
49     pulse = (heat_end - heat_start) / 1000; // calculate pulse width length based on start and end times
50     heat_trig = 2; // indicate internally that the heating pulse was triggered and was finished
51
52     // Serial.print("Start Time: ");
53     // Serial.print(heat_start/1000,3);
54     // Serial.println("ms");
55
56     // Serial.print("End Time: ");
57     // Serial.print(heat_end/1000,3);
58     // Serial.println("ms");
59
60     Serial.print("Heating Time: "); // display heating time in serial
61     Serial.print(pulse, 3);
62     Serial.println(" ms");
63     Serial.println();
64   }
65
66   // if the oscilloscope triggering switch was turned off
67   if (digitalRead(9) == LOW) {
68     osc = 0; // reset oscilloscope triggering
69   }
70
71   // if the heating trigger was pulsed and concluded but the button was not pushed again
72   if (digitalRead(4) == LOW && heat_trig == 2) {
73     heat_trig = 0; // reset heating trigger state
74   }
75 }
76

```

**Figure G-3:** Arduino IDE code used to simultaneously trigger a continuous collection signal and a single shot heating signal.

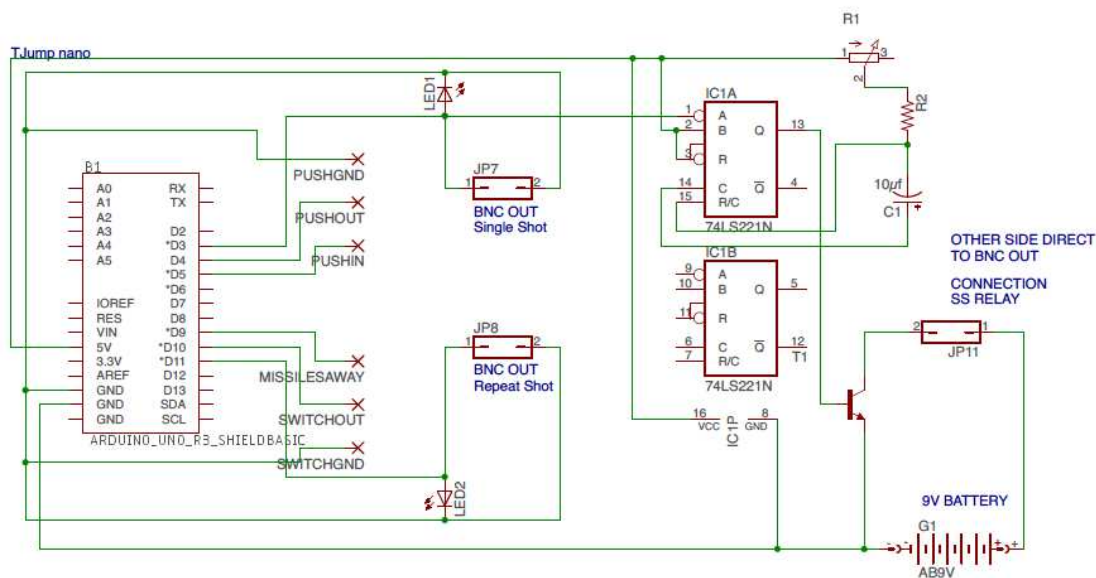
The heating aspect of this circuit utilizes a retriggerable monostable vibrator chip (SN74LS221N) which is powered by the Arduino 5V pin and activated by the programmed Arduino waveform input. The chip takes the input waveform and elongates the pulse width using timing components in the form of a capacitor and potentiometer. The width of the output waveform is related to the discharge time of the

capacitor by the relation  $\tau = 0.7RC$ . Having the resistor be a potentiometer allows one to have relative control over the width of the output waveform. This added component does nothing but modulate the input waveform. In order to gain control over both the output waveform width and amplitude, an amplifying BJT transistor is powered by a 9V battery and actuated by the output of the monostable vibrator chip. This action controls a solid-state relay which is connected in series to a separate circuit loop which initiates heating of a the already existing T-Jump probe by a variable power supply. In this way the heating behavior of the wire can be programmed directly by the Arduino and executed by the monostable vibrator in a variety of ways which can be simultaneously measured by a continuously triggered oscilloscope for a select number of data segments.



**Figure G-4:** Theoretical output signals for heating and Arduino control (BNC outputs).



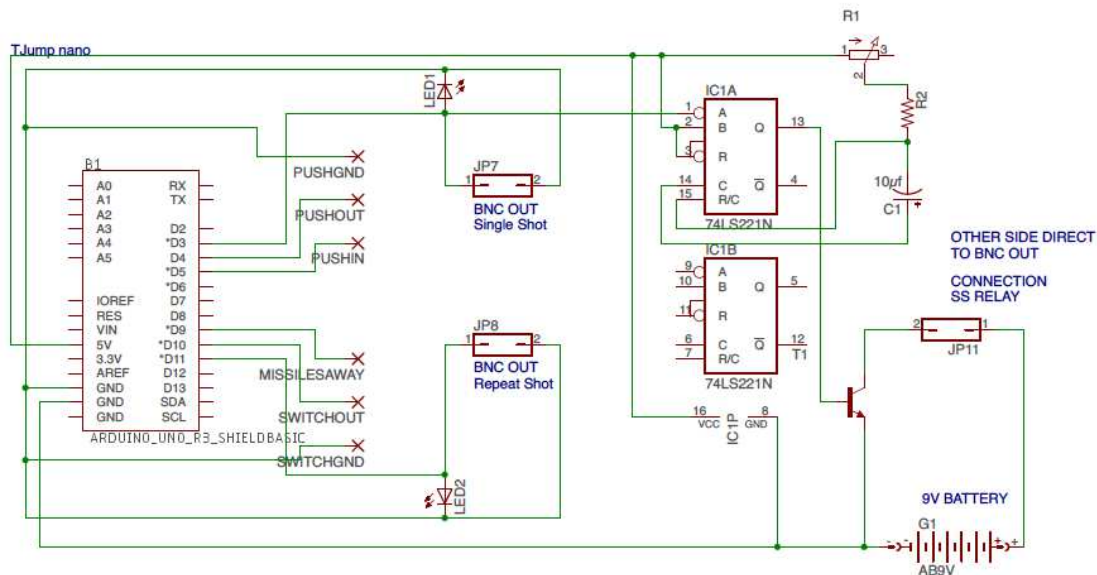


335

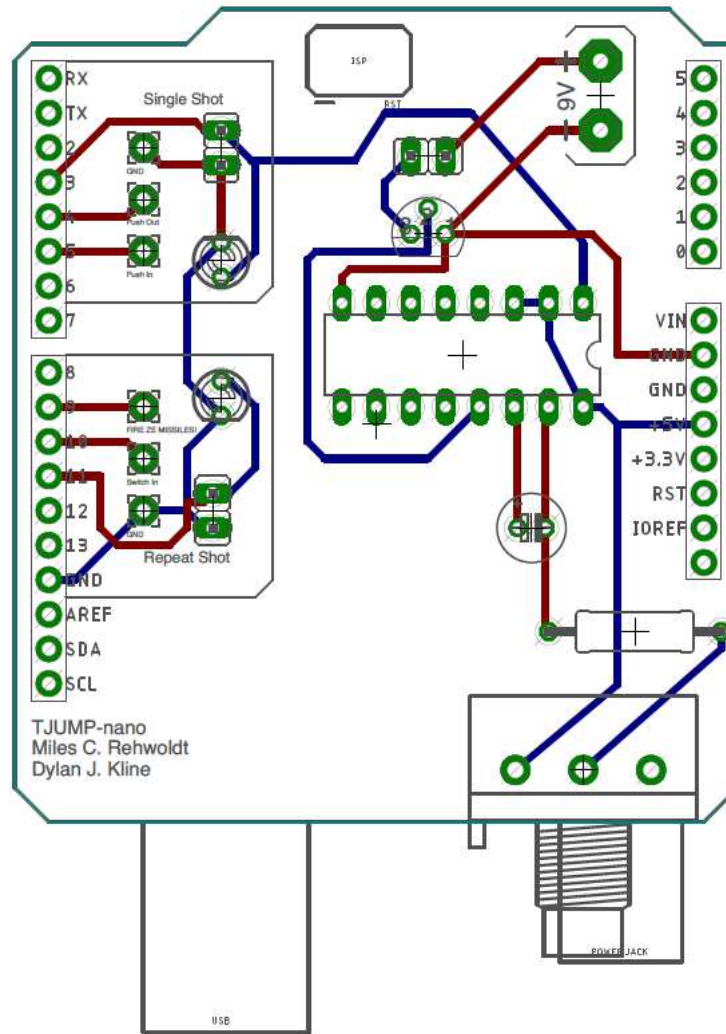


### Printed circuit board design

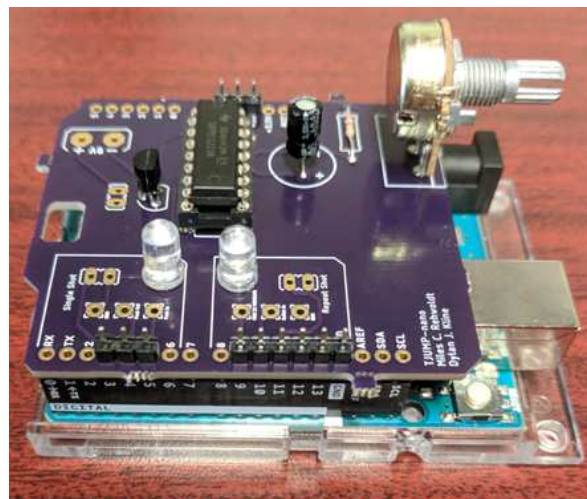
The circuitry of this build was designed so that the corresponding circuit board would attach directly the Arduino as an Arduino shield. This eliminates the use of the bulkier heating pulse box and function generator and replaces them with a roughly 4x4 in board with a 2-inch height. Combined with the solid-state relay and BNC connects, this corresponds to a build which can easily fit inside a small enclosure that is several times smaller than the combined real estate taken up by its predecessors. Although the Arduino does need a computer to initially program and power it, this responsibility is simply taken up by the oscilloscope which itself is a windows computer acting as a consolidated power source and programming interface. Below, Figures G6-G9 illustrate the circuit scheme, PCB layout and Arduino shield, and the fabricated aluminum enclosure.



**Figure G-6:** Schematic of T-Jump nano circuit board as drawn in Autodesk Eagle 8.4.2.



**Figure G-7:** Diagram of custom printed circuit board to serve as shield for Arduino Uno (Rev3). Printed circuit board is manufactured by OSH Park. Red traces are printed on top of the shield and blue traces are printed on the bottom of the shield.



**Figure G-8:** PCB of T-Jump Nano circuit mounted to Arduino Uno board.



**Figure G-9:** PCB aluminum enclosure with output switches, BNC connectors and heating relay.

#### I/O configuration for T-Jump Nano printed circuit board

The Arduino Uno circuit board used in the T-Jump Nano has a variety of digital pins that are programmed for signaling upon certain events such as pressing a button or flipping a switch. The table below can be used as a troubleshooting or programming guide for future reference by others.

**Table G-1:** Arduino input/output pin specification for T-Jump Nano.

Pin	Setting	Input	Output
D3	OUTPUT		Pin 1A on SN74LS221N Single shot BNC
D4	INPUT	D5, push button	
D5	OUTPUT HIGH		Push button to D4
D9	INPUT	D10, switch	

D10	OUTPUT HIGH	Switch to D9
D11	OUTPUT	Repeat shot BNC

#### ***G.4. Other uses***

The T-Jump Nano has been further adapted as a resource for multiple signal generation and TTL signaling in multiple experiments across the Zachariah group. Custom routines and experimental descriptions for use with the time of flight mass spectrometer, pressure cell, high speed microscope, spatially focused microwave, and general TTL signaling can be seen in the following appendix.

#### ***G.5. Acknowledgements***

This work was done as part of the final project for PHYS 685 Research Electronics at the University of Maryland College Park in the Fall 2018 semester. Dylan J. Kline and Miles C. Rehwoldt contributed equally to this work.

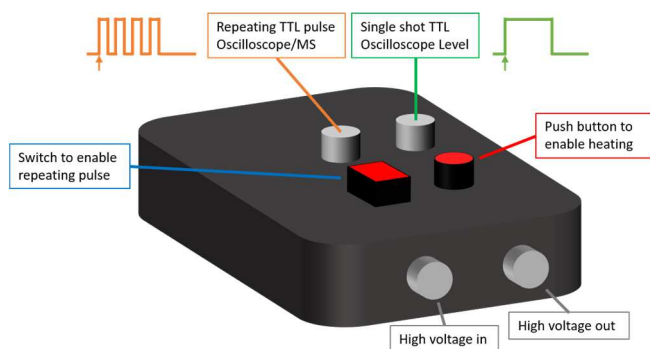
## Appendix H: Arduino scripts for the T-Jump Nano in different configurations

As discussed in the previous appendix, the T-Jump nano uses an Arduino Uno as the basis for generating the signals that simultaneously trigger a variety of different instruments. Numerous programs have been written to accompany the T-Jump nano in its original form and have been included here for reference in the future.

\*\*\*DISCONNECT ALL PERIPHERALS BEFORE CHANGING THE CODE AND UPLOADING IT TO THE BOARD.\*\*\*

### ***H.1. T-Jump Nano Time of Flight Mass Spectrometry (TOFMS)***

The original intention of the T-Jump was to reduce the footprint of the large desk T-Jump apparatus that has been a part of the lab and make the device more versatile. This program outputs a solid TTL signal to trigger to a camera, a repeating TTL signal at desired intervals to trigger data collection on an oscilloscope, and allows for current to flow through a wire with very high timing accuracy. In order to activate the repeating signals to the oscilloscope, the red missile switch must be turned to the ON position (lit up). More details on this setup can be seen in the previous appendix and a typical setup can be seen in Figure H-1. Pushing down on the red button will allow for current to flow through the solid-state relay, will trigger the camera, and will send repeating pulses to the oscilloscope the desired frequency.



**Figure H-1:** Example setup for T-Jump Nano for use with high-speed time of flight mass spectrometer.

```

void setup() {
    pinMode(5, OUTPUT); // input to push button for single shot
    pinMode(4, INPUT);  // output to push button for single shot
    pinMode(3, OUTPUT); // signal sent to single shot segment of
circuit
    pinMode(11, OUTPUT); // signal sent to repeating pulses
    pinMode(10, OUTPUT); // input to switch for repeating pulses
    pinMode(9, INPUT);   // output to switch for repeating pulses
    Serial.begin(250000); //
}

void loop() {

    static int heat_trig = 0; // pre (0), during (1), or
post (2) trigger
    static int osc = 0; // determines whether or not
to send trigger to oscilloscope

    unsigned long heat_time = 5000; // total time for heating in
microseconds
    static unsigned long heat_start; // stored value for heating
start time
    unsigned long heat_end; // stored value for heating
finish time
    float pulse; // value calculated to
determine pulse length
    static unsigned long current; // stored value for current
time in loop

    digitalWrite(5, HIGH); // sets input of trigger for
heating to high
    digitalWrite(10, HIGH); // sets input of
oscilloscope to high

    // if the heating trigger has been pushed and the trigger was
reset
    if (heat_trig == 0 && digitalRead(4) == HIGH) {
        osc = 1; // turn on oscilloscope
        triggering signal
        heat_trig = 1; // indicate internally that
the heating trigger was pushed
    }
}

```

```

        heat_start = micros();                // record the time right
before heating in microseconds
        digitalWrite(3, HIGH);                // start heating
    }

    // if the trigger has been pushed before and the oscilloscope
switch is open
    if (osc == 1 && digitalRead(9) == HIGH) {
        digitalWrite(11, HIGH);              // send pulse to
oscilloscope
        delayMicroseconds(100);              // delay execution for 20us
        digitalWrite(11, LOW);               // stop pulse to
oscilloscope
    }

    // if the trigger was pushed, continually record the time
    if (heat_trig == 1) {
        current = micros();
    }

    // if the time the wire has been heating is equal to or greater
than the desired time
    if ((current - heat_start) >= heat_time && heat_trig == 1) {
        digitalWrite(3, LOW);                // stop heating
        //heat_end = current;                // record time at which
the wire stopped heating
        //pulse = (heat_end - heat_start) / 1000; // calculate pulse
width length based on start and end times
        heat_trig = 2;                      // indicate internally that
the heating pulse was triggered and was finished

        //    Serial.print("Start Time: ");
        //    Serial.print(heat_start/1000,3);
        //    Serial.println("ms");

        //    Serial.print("End Time: ");
        //    Serial.print(heat_end/1000,3);
        //    Serial.println("ms");

        //    Serial.print("Heating Time: ");    // display heating
time in serial
        //    Serial.print(pulse, 3);
        //    Serial.println(" ms");
        //    Serial.println();
    }

    // if the oscilloscope triggering switch was turned off
    if (digitalRead(9) == LOW) {
        osc = 0;                            // reset oscilloscope
triggering
    }

    // if the heating trigger was pulsed and concluded but the button
was not pushed again
    if (digitalRead(4) == LOW && heat_trig == 2) {

```

```

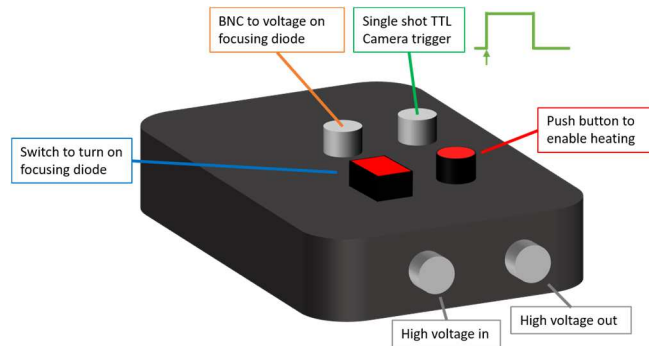
        heat_trig = 0;                                // reset heating trigger
    state
    }

}

```

## H.2. T-Jump Nano Microscope

A variant of the T-Jump nano code that allows for an external device (typically a light) to be connected to the second (repeating) TTL output as shown in Figure H-2 which can be turned on/off using the red missile switch. The remaining TTL output is for triggering a camera and the high voltage throughputs operate normally. Pushing down on the red button will allow for current to flow through the solid-state relay and will trigger the camera. Generally, the red button is held down until an ignition event is observed.



**Figure H-2:** Example setup for T-Jump Nano for use with high-speed microscope or microwave needle apparatus.

```

void setup() {
    pinMode(5, OUTPUT); // input to push button for single shot
    pinMode(4, INPUT);  // output to push button for single shot
    pinMode(3, OUTPUT); // signal sent to single shot segment of
circuit
    pinMode(11, OUTPUT); // signal sent to repeating pulses
    pinMode(10, OUTPUT); // input to switch for repeating pulses
    pinMode(9, INPUT);   // output to switch for repeating pulses
    Serial.begin(250000); //
}

// special modification for microscope experiment

void loop() {

```



```

    static int heat_trig = 0;                // pre (0), during (1), or
post (2) trigger
    static int osc = 0;                     // determines whether or not
to send trigger to oscilloscope

    unsigned long heat_time = 500000;       // total time for heating in
microseconds
    static unsigned long heat_start;         // stored value for heating
start time
    unsigned long heat_end;                 // stored value for heating
finish time
    float pulse;                           // value calculated to
determine pulse length
    static unsigned long current;           // stored value for current
time in loop

    digitalWrite(5, HIGH);                  // sets input of trigger for
heating to high
    digitalWrite(10, HIGH);                 // sets input of
oscilloscope to high

    // CONTROL DIODE
    if (digitalRead(9) == HIGH) {
        digitalWrite(11, HIGH);             // turn on diode for
focusing
    }
    else
    {
        digitalWrite(11, LOW);              // turn off diode for
focusing
    }
    // END DIODE CONTROL

    // if the trigger has been pushed before and the oscilloscope
switch is open
    if (digitalRead(4) == HIGH) {
        heat_trig = 1;
        osc = 1;
        digitalWrite(3, HIGH);              // send pulse to oscilloscope
    }

    // if the trigger was pushed, continually record the time
    if (heat_trig == 1) {
        current = micros();
    }

    // if the time the wire has been heating is equal to or greater
than the desired time
    if ((current - heat_start) >= heat_time && heat_trig == 1) {
        digitalWrite(3, LOW);               // stop heating
        heat_trig = 2;                      // indicate internally that
the heating pulse was triggered and was finished
    }

    // if the oscilloscope triggering switch was turned off
    if (digitalRead(9) == LOW) {

```

```

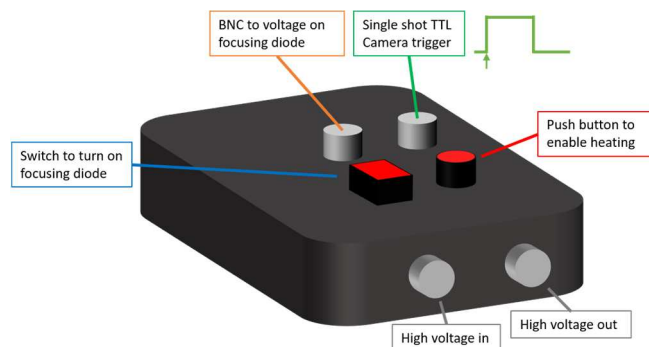
    osc = 0;                                // reset oscilloscope
    triggering
  }

  // if the heating trigger was pulsed and concluded but the button
  was not pushed again
  if (digitalRead(4) == LOW && heat_trig == 2) {
    heat_trig = 0;                          // reset heating trigger
    state
  }
}

```

### ***H.3. T-Jump Nano for Pressure Cell and Spectrometry Measurements***

A variant of the T-Jump nano code that can be used for tests involving the pressure cell that also involve data collection for high-speed spectroscopy. The required setup can be seen in Figure H-3. Most importantly, the signal being sent to the 32-channel spectrometer must be on the TTL output that is typically reserved for repeating signals and the code has been modified to instead put out a stable signal on that channel. Failing to connect to the right TTL channel will result in abnormal data spacing on the collection for the high-speed spectrometer. Pushing down on the red button will allow for current to flow through the solid-state relay and will trigger the oscilloscope/spectrometer.



**Figure H-3:** Example setup for T-Jump Nano for use with high-speed spectrometry and pressure cell measurements.

```

void setup() {
  pinMode(5, OUTPUT); // input to push button for single shot
  pinMode(4, INPUT);  // output to push button for single shot
}

```

```

    pinMode(3, OUTPUT); // signal sent to single shot segment of
circuit
    pinMode(11, OUTPUT); // signal sent to repeating pulses
    pinMode(10, OUTPUT); // input to switch for repeating pulses
    pinMode(9, INPUT); // output to switch for repeating pulses
    Serial.begin(250000); //
}

// special modification for pressure cell experiment
// before, we were having issues communicating data to the high-
speed vertilon spectrometer because it only reads when the input
voltage is high...
// with the previous setup, this made things pretty unfortunate when
trying to record at 400kHz in 200us increments (the way the old code
was set up was not ideal).
// so this has been written in it's place. here's hoping it works.
// if the pulse width is not long enough for whatever reason, then
we can change it programatically.

void loop() {

    static int heat_trig = 0; // pre (0), during (1), or
post (2) trigger
    static unsigned long heat_time = 500000; // total time for
heating in milliseconds
    static unsigned long heat_start; // stored value for heating
start time
    static unsigned long current; // stored value for current
time in loop

    digitalWrite(5, HIGH); // sets input of trigger for
heating to high
    digitalWrite(10, HIGH); // sets input of
oscilloscope to high

    // if the trigger has been pushed before and the oscilloscope
switch is open
    if (digitalRead(4) == HIGH) {

        heat_trig = 1; // change triggering status
to "pushed"
        digitalWrite(11, HIGH); // send pulse to
oscilloscope
        digitalWrite(3, HIGH); // send pulse to
oscilloscope

    }

    // if the trigger was pushed, continually record the time
    if (heat_trig == 1) {
        current = micros();
    }

    // if the time the wire has been heating is equal to or greater
than the desired time

```

```

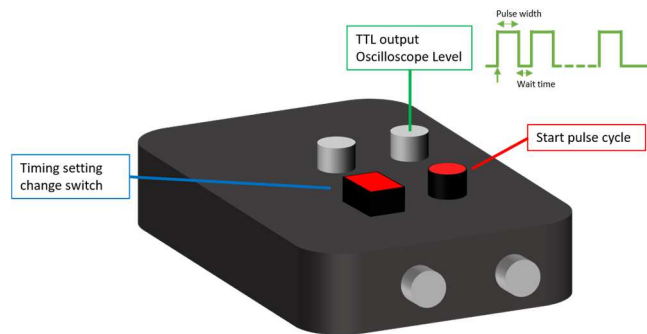
    if ((current - heat_start) >= heat_time && heat_trig == 1) {
        digitalWrite(3, LOW);           // stop heating
        heat_trig = 2;                 // indicate internally that
        the heating pulse was triggered and was finished
    }

    // if the heating trigger was pulsed and concluded but the button
    was not pushed again
    if (digitalRead(4) == LOW && heat_trig == 2) {
        digitalWrite(11, LOW);         // stop pulse to
        oscilloscope
        heat_trig = 0;                 // reset heating trigger
        state
    }
}

```

#### H.4. T-Jump Nano TTL

A variant of the T-Jump nano code that is not designed for heating purposes, but rather for TTL signaling (such as with a camera or laser). This code gives users the option to change the pulse length, width between pulses, and number of pulses for a single run. A typical setup for this use case can be seen in Figure H-4.



**Figure H-4:** Example setup for T-Jump Nano for use as a TTL signaling device with custom number of pulses, wait times, and pulse widths.

```

void setup() {
    // Single shot, standard push button
    pinMode(5, OUTPUT); // input to push button for single shot
    pinMode(4, INPUT);  // output to push button for single shot
    pinMode(3, OUTPUT); // signal sent to single shot segment of
    circuit

    // Missile switch, modified scenario
    pinMode(10, OUTPUT); // output to switch for longer pulses
    pinMode(9, INPUT);  // input to switch for longer pulses
}

```

```

digitalWrite(5, HIGH); // write pin 5 high
digitalWrite(10, HIGH); // write pin 9 high

Serial.begin(25000); //
}

void loop() {

    // VARIABLE DECLARATION
    static int single_trig = 0;           // pre (0), during (1), or
post (2) trigger
    static long TTL_time;

    // CHANGE TIMES
    TTL_time=1000;                       // pulse time with switch
down in milliseconds
    if (digitalRead(9)==HIGH) TTL_time=1000; // pulse time with switch
up in milliseconds

    // CHANGE REPEATS AND SPACING
    unsigned long wait_time = 0;        // total waiting time
    static int repeat = 1;              // repeated shot

    // MAIN FUNCTION
    // if the heating trigger has been pushed and the trigger was
reset
    if (single_trig == 0 && digitalRead(4) == HIGH) {
        single_trig = 1;                // indicate internally
that the heating trigger was pushed

        for (int i = 1; i <= repeat; i++) {
            digitalWrite(3, HIGH);      // start heating
            delay(TTL_time);             // start pulse
            digitalWrite(3, LOW);        // end pulse, begin wait
            delay(wait_time);            // wait
        }

    }

    // RESET
    // if the heating trigger was pulsed and concluded but the button
was not pushed again
    if (digitalRead(4) == LOW && single_trig == 1) {
        single_trig = 0;                // reset single trigger
state
    }

}

```

## Appendix I: T-Jump Nano Ultimate – A better multi-signal pulse generator for precision triggering and heating

### ***1.1. Introduction***

A cornerstone instrument in the Zachariah Group is what is known as the Temperature Jump (T-Jump) Time of Flight Mass Spectrometer (TOFMS) and an improved signal generator was designed in late 2017 to consolidate space and increase versatility in the electrical heating element of the device (see Appendix G). Since the T-Jump Nano did not have a graphical user interface (GUI), it was difficult for users that were not adept to programming to make changes to the code without detailed knowledge of the device itself and it was an inherent danger if an unsuspecting user changed the program between experiments. As a solution to this, the T-Jump Nano Ultimate was designed to implement a GUI (screen readout) with a human-machine interface (joystick) to alleviate the need for detailed programming knowledge of the device. The remaining capabilities of the device are identical to those of the T-Jump Nano and can be seen in Appendix G.

### ***1.2. Project purpose***

The purpose of this project is to improve usability of the T-Jump Nano by implementing a screen readout and joystick for manual manipulation of heating curves/TTL signals without detailed programming knowledge of the device. There is nothing inherently wrong with the T-Jump Nano, but this serves as an improvement over the current model.

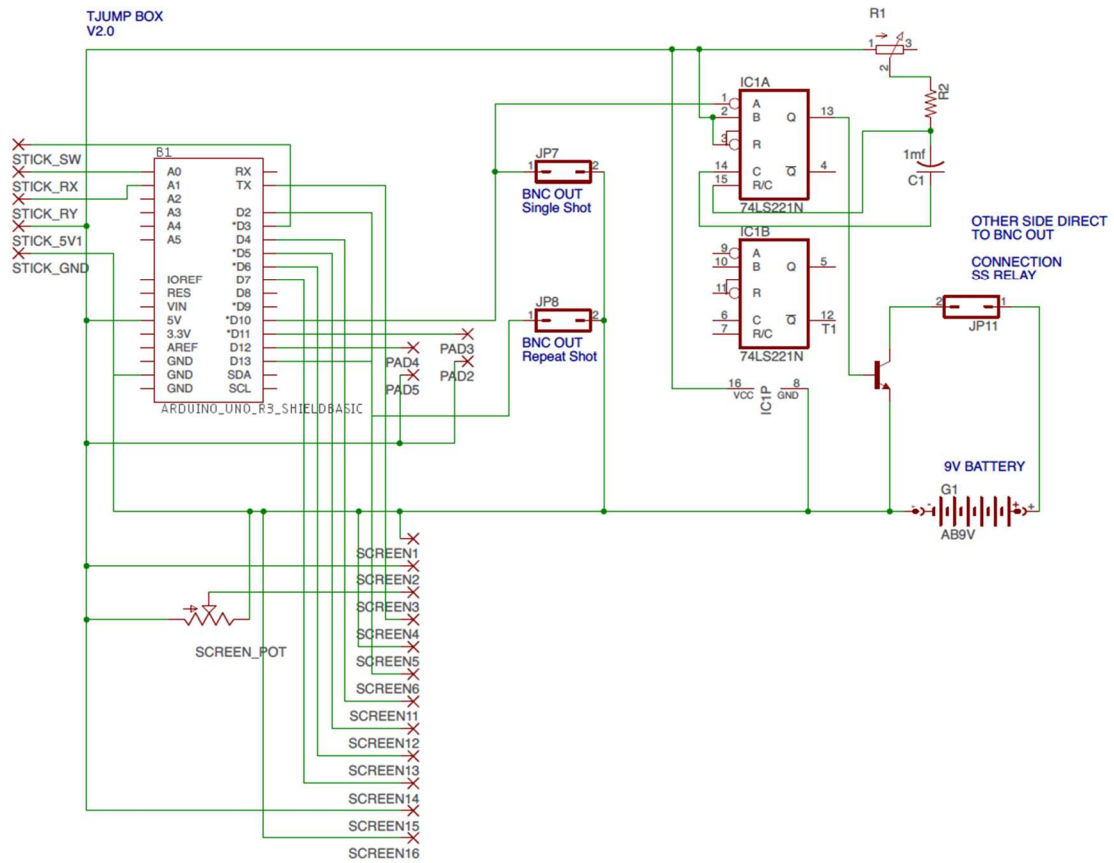
### ***1.3. T-Jump Nano Ultimate***

The T-Jump Nano Ultimate's primary heating and triggering mechanism is identical to that of the original T-Jump Nano as detailed in Appendix G, however some of the input and output options have changed to accommodate a new LCD screen readout and joystick. As a result, only the printed circuit board design and I/O description is provided in this Appendix and it is left to the user to reference Appendix G for the theory behind the T-Jump and T-Jump Nano operation.

#### **Printed circuit board design**

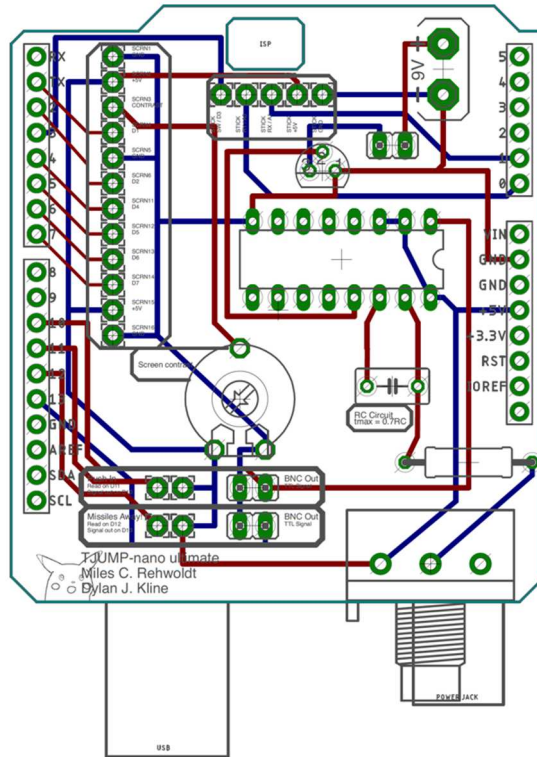
A custom printed circuit board was designed to attached directly to an Arduino Uno as an Arduino Shield to eliminate use of bulkier equipment and the solderless breadboard. The final breadboard is designed to have two BNC outputs which represent different output options for TTL signaling/triggering, an LCD screen attached with standard square female-female cables, and a joystick that is also to be attached with standard square female-female cables. The overall size of the final breadboard remains ~4x4 in with a 2-inch height for materials soldered directly to the breadboard. The entire build remains small enough to fit inside a small enclosure similar to the size of the original T-Jump Nano. In principle, users would no longer need to program the Arduino board as frequently given the new functionality for changing the program with the joystick and reading the triggering/heating scheme with the LCD screen, though a computer is still needed to power the board and load the initial program. For later reference, a 10 $\mu$ F capacitor, 150 k $\Omega$  variable resistor, and 200  $\Omega$  resistor are used for the heating circuit.

Below, Figures I1-I3 illustrate the circuit scheme, PCB layout and Arduino shield.

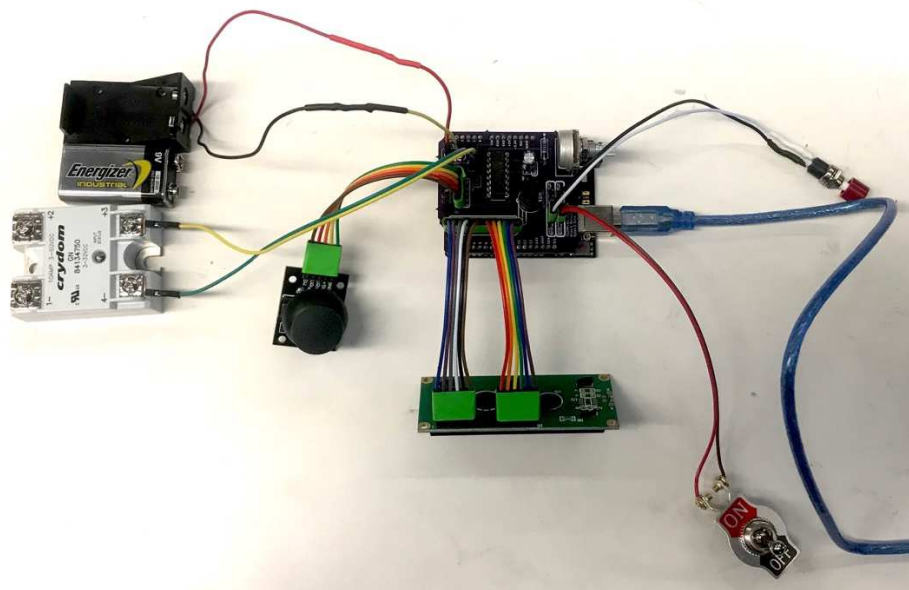


**Figure I-1:** Schematic of T-Jump Nano Ultimate circuit board as drawn in Autodesk Eagle 8.4.2.





**Figure I-2:** Diagram of custom printed circuit board to serve as shield for Arduino Uno (Rev3). Printed circuit board is manufactured by OSH Park. Red traces are printed on top of the shield and blue traces are printed on the bottom of the shield.



**Figure I-3:** PCB of T-Jump Nano Ultimate circuit mounted to Arduino Uno board.

## I/O configuration for T-Jump Nano printed circuit board

The Arduino Uno circuit board used in the T-Jump Nano Ultimate has a variety of digital pins that are programmed for signaling upon certain events such as pressing a button or flipping a switch. The table below can be used as a troubleshooting or programming guide for future reference by others.

**Table I-1:** Arduino input/output pin specification for T-Jump Nano Ultimate.

Pin	Setting	Input	Output
A0	INPUT	Joystick	Joystick RY
A1	INPUT	Joystick	Joystick RX
D1	OUTPUT		Screen pin 4
D2	OUTPUT		Screen pin 6
D3	INPUT	Joystick	Joystick SW
D4	Output		Screen pin 11
D5	OUTPUT		Screen pin 12
D6	OUTPUT		Screen pin 13
D7	OUTPUT		Screen pin 14
D10	OUTPUT		Pin 1A on SN74LS221N Single shot BNC
D11	INPUT	push button, 5V	
D12	INPUT	switch, 5V	
D13	OUTPUT HIGH		Repeat shot BNC

### G.5. Acknowledgements

The initial work for the T-Jump Nano was done as part of the final project for PHYS 685 Research Electronics at the University of Maryland College Park in the Fall 2018 semester. This updated circuit board was designed and assembled in Winter 2019 by Dylan J. Kline and Miles C. Rewholdt to fit the needs of the group.

## Appendix J: Arduino scripts for the T-Jump Nano Ultimate

As discussed in the previous appendix, the T-Jump Nano Ultimate uses an Arduino Uno as the basis for generating the signals that simultaneously trigger a variety of different instruments. Unlike the T-Jump Nano, however, only one program has been written for this device, though principles from scripts of the T-Jump Nano could be easily ported over for use with the T-Jump Nano Ultimate. The code for the T-Jump Nano Ultimate is provided below.

\*\*\*DISCONNECT ALL PERIPHERALS BEFORE CHANGING THE CODE AND  
UPLOADING IT TO THE BOARD.\*\*\*

```
#include <LiquidCrystal.h>
#include "IRremote.h"

LiquidCrystal lcd(1, 2, 4, 5, 6, 7); // pins for lcd display

#define debounce 20 // ms debounce period to prevent flickering when
pressing or releasing the button
#define holdTime 500 // ms hold period: how long to wait for
press+hold event

// variable declaration
static int    heat_trig = 0;          // status (0) = not triggered or
reset, (1) = triggered
unsigned long heat_time    = 1000; // pulse width in ms
unsigned long wait_time    = 2000; // wait time in ms
unsigned int  repeat       = 3;      // number of repeats. 1 = one run
static int    pw_print     = 0;      // printing status
static unsigned long heat_start;      // stored value for heating start
time
static unsigned long current;          // stored value for current time
in loop

//Joystick pin input/output
int xPin = A0;          //X-axis input pin
int yPin = A1;          //Y-axis input pin
int buttonPin = 3;      //Button input pin
static int xPosition = 0; //Initial x-axis joystick position
static int yPosition = 0; //Initial y-axis joystick position

// Button variables
```

```

int buttonVal = 0;           // value read from button
int buttonLast = 0;         // buffered value of the button's previous
state
long btnDnTime;             // time the button was pressed down
long btnUpTime;             // time the button was released
boolean ignoreUp = false;   // whether to ignore the button release
because the click+hold was triggered

// Setup of time stepping and the likes
int tstepp = 50;            // ms step size for the joystick
int tstepw = 100;           // ms step size for the joystick
int tdel = 250;             //ms delay when updating screen
long joyDnTime;             // future implementation of ramping

void setup() {
    // INFO FOR LCD SCREEN
    lcd.begin(16,2);         // inintializing type of LCD
screen
    pinMode(xPin, INPUT);    // x-axis input
    pinMode(yPin, INPUT);    // y-axis input
    pinMode(buttonPin, INPUT_PULLUP); // button toggle
    digitalWrite(buttonPin, HIGH ); // button input
    lcd.clear();

    pinMode(11, INPUT);      // signal input from heating trigger
    pinMode(10, OUTPUT);     // signal input from heating trigger
}

void loop(){
    xPosition = analogRead(xPin); // Reads x-axis input
    yPosition = analogRead(yPin); // Reads y-axis input
    buttonVal = digitalRead(buttonPin); // Reads toggle button input

    // Joystick X-Axis: Controls heat_time
    if (yPosition >= 450 && yPosition <= 550) {
        if(xPosition < 400)
        {
            if(heat_time <= 0)
            {
                heat_time = 0;
            }
            else
            {
                heat_time -= tstepp;
                delay(tdel);
            }
        }
        else if(xPosition > 600)
        {
            heat_time += tstepp;
            delay(tdel);
        }
    }
}

```

```

// Joystick Y-Axis: Controls wait_time
if (xPosition >= 450 && xPosition <= 550) {
    if(yPosition < 400)
    {
        wait_time += tstepw;
        delay(tdel);
    }
    else if(yPosition > 600)
    {
        if(wait_time <= 0)
        {
            wait_time = 0;
        }
        else
        {
            wait_time -= tstepw;
            delay(tdel);
        }
    }
}

// Test for button pressed and store the down time
if (buttonVal == LOW && buttonLast == HIGH && (millis() -
btnUpTime) > long(debounce))
{
    btnDnTime = millis();
}

// Test for button release and store the up time
if (buttonVal == HIGH && buttonLast == LOW && (millis() -
btnDnTime) > long(debounce))
{
    if (ignoreUp == false) event1();
    else ignoreUp = false;
    btnUpTime = millis();
}

// Test for button held down for longer than the hold time
if (buttonVal == LOW && (millis() - btnDnTime) > long(holdTime))
{
    event2();
    ignoreUp = true;
    btnDnTime = millis();
}

buttonLast = buttonVal;

// printing information to screen
lcd.setCursor(0,0);
lcd.print("P:");           // Pulse width
lcd.print(heat_time);      // TTL time
lcd.print("ms");           // spacer

lcd.print("  ");

```

```

    lcd.print("x");
    lcd.print(repeat);          // repeats
    lcd.print("  ");

    lcd.setCursor(0,2);        // write "running" to screen
    lcd.print("W:");           // Pulse width
    lcd.print(wait_time);      // wait time
    lcd.print("ms");           // spacer
    lcd.print("  ");

    pw_print = 1;              // save info to screen w/o updating

    // if trigger is pushed and the system was reset
    if (heat_trig == 0 && digitalRead(11) == HIGH){
        heat_trig = 1;         // status update for trigger
        lcd.setCursor(0,2);     // write "running" to screen
        lcd.print("RUNNING! #"); // write "running" to screen

        for (int i = 1; i <= repeat; i++){ // loop for sending ttl
signals
            lcd.setCursor(10,2);          // write the run it is
currently on
            lcd.print(i);                 // write the run it is
currently on
            digitalWrite(10, HIGH);        // 5V to destination
            delay(heat_time);              // wait for TTL time
            digitalWrite(10, LOW);         // 0V to destination
            delay(wait_time);              // delay the wait time as
specified
        }

    }

    // status reset
    if (digitalRead(11) == LOW && heat_trig ==1){
        heat_trig = 0;                 // status reset
        lcd.clear();                   // clear screen
        lcd.print("DONE :)");          // indicate done
        delay(1000);                   // delay
        lcd.clear();                   // clear
        pw_print = 0;                   // reset screen
    }
}

//=====
// Events to trigger by click and press+hold
void event1()
{
    repeat++;
    lcd.setCursor(9,0);
    lcd.print("x");
    lcd.print(repeat);                // repeats
}

void event2()

```

```
{
  if(repeat <= 0)
  {
    repeat = 0;
  }
  else
  {
    repeat--;
  }

  lcd.setCursor(9,0);
  lcd.print("x");
  lcd.print(repeat);           // repeats
}
```

## Appendix K: Spatially focused microwave needle apparatus operation

### *K.1. Introduction, project purpose, and background*

This section will serve as a user manual for the spatially focused microwave needle apparatus that was designed for the research performed in Chapter 8. As part of the effort to spatially localize ignition events of energetic materials using microwave-frequency oscillations in the electromagnetic field, it was necessary to design a device that would be able to generate a single frequency chirp with enough power to heat and/or ignite samples with the option to change the signal frequency and power output. The apparatus designed consists of a DS Instruments SG12000 which can generate signals with frequencies ranging from 25-12000 MHz and with a maximum power output of 10 dBm (10 mW). The signal is then routed into a solid-state AMCOM AM206545SF-3H microwave amplifier (operating frequency 2-6 GHz) which has can provide a ~32 dB gain to the incoming signal when power is applied, bringing the total signal power output to a maximum of ~15 W. This amplified signal is then ported out through a coaxial cable with terminated with an SMA female-to-solder cup connector, to which a metal needle is soldered into the solder cup. The needle can be of any length or diameter, but for the experiments performed in Chapter 8, a simple sewing needle (Singer) was used. It should be noted here that while the maximum gain can result in a ~15 W signal, the actual operating gain of the amplifier may be less since the needle is not a well-tuned antenna.



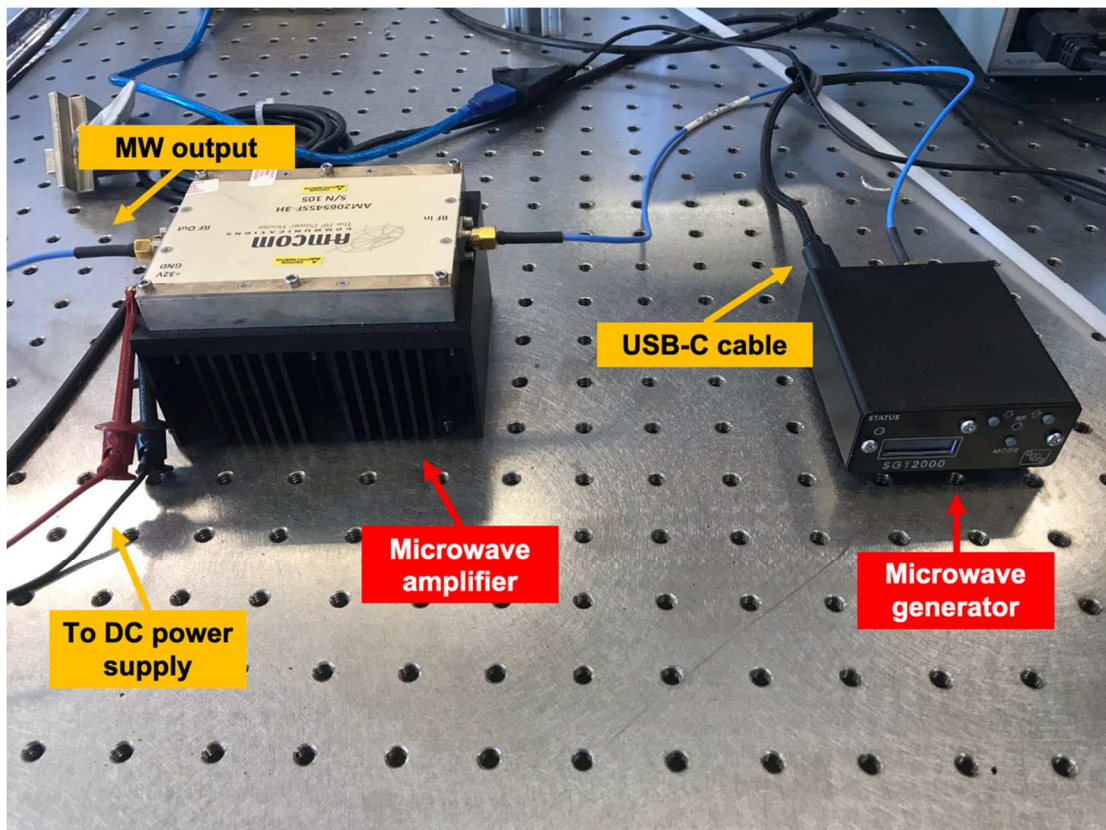
As also discussed in Chapter 8, the sample exposure is actually in the “near-field” of the electromagnetic (EM) field, implying there will be some coupling between sample and antenna. This is because the working distance from emitter to sample is much smaller than the operating wavelength. Samples are not being exposed to radiating microwave energy, but rather they are being exposed to local oscillations in the electromagnetic field at those microwave frequencies. The resulting phenomena is the same as if the samples were being exposed to similar-frequency radiation since radiation implies that the wave is traveling through space from a distance far away from the emitter, whereas this is just local oscillations. As an analogy, if you are in a wave pool (popular in the early-to-mid 2000’s), the change in water level (or the wave) is akin to the electromagnetic oscillation and radiation is the translation of the wave down the length of the pool. Everybody in the pool experiences the wave (electromagnetic oscillation) all the same since the same wave is traveling down the length of the pool (radiation), just those further away experience the wave a little later in time and perhaps with a little less intensity.

### ***K.2. Microwave needle operation***

Operation of the microwave needle apparatus is relatively simple and is only provided here as a reference. To power the microwave generator, users will need to secure a power supply able to generate a DC power of at least 32 V and 3.3 A continuously.

DO NOT SUPPLY POWER TO THE AMPLIFIER FOR EXTENDED PERIODS OF TIME – DOING SO WILL RESULT IN HIGH TEMPERATURE AND INSTRUMENT DAMAGE.

1. Retrieve DS Instruments SG12000 microwave generator and plug in USB-C power cable. Go through the settings on the microwave generator to manually set the frequency and power. If available, a computer can also be used to control the frequency, power, and even perform frequency sweeps on the microwave generator using the DS Instruments control software.
2. Retrieve solid state power amplifier. Using two SMA cables, attach one cable between the microwave generator and the amplifier input, and another cable should be attached between the output of the amplifier and the desired SMA attachment (e.g. SMA female-to-solder cup with needle). An example setup can be seen in Figure K-1.



**Figure K-1:** Example setup for the microwave generator (DS Instruments) and microwave power amplifier (AMCOM).

3. Using a BNC-to-grabber cable, attach the grabbers to the power inputs of the amplifier. Connect the BNC to the desired power source for the amplifier.

If using the T-Jump Nano to regulate power to the amplifier, use the code for pressure cell/spectrometry measurements (follow the schematic in Figure H-4 of this dissertation as a reference). Power on the power supply to the desired voltage/current. Pressing the red button on the T-Jump Nano will start the application of power to the amplifier. This is the preferred setup since it allows for controlled power supply times and simultaneous triggering with other equipment (such as a camera).

If using a direct connection to the power source for the microwave amplifier, only power on the power supply when samples are ready to be ignited. Once the experiment is over, turn off the power supply.

4. Align microwave applicator with desired sample and apply power to the amplifier to begin experiment. DO NOT SUPPLY POWER TO THE AMPLIFIER FOR EXTENDED PERIODS OF TIME – DOING SO WILL RESULT IN HIGH TEMPERATURE AND INSTRUMENT DAMAGE.
5. Once experiments are complete, be sure to unplug the microwave generator and turn off power supply to amplifier.

### ***K.3. Acknowledgements***

This apparatus was designed largely in part with Charles J. Turner and his input at every step of the design process, including performing microwave simulations to

verify proper operating principles, is greatly appreciated. Publications that use this apparatus should refer to D.J. Kline\*, M.C. Rehwoldt\*, C.J. Turner, P. Biswas, G.W. Mulholland, S.M. McDonnell, M.R. Zachariah, Spatially focused microwave ignition of metallized energetic materials, J. Appl. Phys. 127 (2020) 55901. doi:10.1063/1.5134089.

## Appendix L: Thermophysical property estimation codes for steadily-propagating, gasless reacting materials

This appendix will serve as the user manual for the thermal property estimation code as has been employed in the research originally described in Chapter 6. The code was designed as a use for high-speed videography experiments where temperature of the reaction can provide useful insight on the mechanisms or heat transfer rates by estimating the thermal diffusivity of a sample or activation energy of reaction. This code was originally written to analyze samples provided by Draper Laboratory that were gasless SHS samples, but could be applied in other areas if necessary. This code requires data that is output from the pyrometry code that is discussed in Appendix D. Throughout the manual, text in `this font` will represent variables or expressions that are used in a MATLAB command.

The manual below is broken up into different sections on (1) running the thermal property estimation code, (2) input and output information for the program, (3) detailed variable listing and descriptions, (4) generalized code structure, (5) a summary, (6) version history notes if applicable, and (7) the raw MATLAB code.

### ***L.1. Running thermophysical property estimation code***

The executable MATLAB script can be found in the Zachariah Group Drive at Volumes/users/Kline/Codes/Thermal-Property-Pyrometry-Estimation/thermal\_property\_estimator\_7.m. An online version has also been stored with version control and is available by reasonable request. The code has numerous dependencies on installed packages for MATLAB and those that have been written by

the author and stored in Volumes/users/Kline/Codes/Thermal-Property-Pyrometry-Estimator/.

**Table L-1:** Dependencies for thermal property estimation code.

<b>Name</b>	<b>Description</b>
MATLAB Image Processing Toolbox	General use toolbox which contains required functions for image processing related purposes.
MATLAB Curve Fitting Toolbox	General use toolbox which contains required functions for curve fitting to estimate thermal diffusivity.
polyfix	Used to create a polynomial fit with fixed parameters like derivatives at certain points. Provided in external functions but available for download on the MathWorks website.
smooth2a	Used to smooth 3D data set in two dimensions. Provided in external functions but available for download on the MathWorks website.
hline_vline	Inserts vertical lines on plots to show where regions of interest are for estimating alpha.

The opening lines of thermal\_property\_estimator\_7.m attempts to automatically add the required external functions but if this fails it can always be done manually by right-clicking on the External Functions folder and choosing Add to Path>Selected Folder and Subfolders.

To run kline\_temperature\_colorcam10.m, open the folder in the script viewer in MATLAB and click Run in the Editor Ribbon. The program will execute automatically from the first written line. Alternatively, users can choose to Run Section in the script viewing window since the code has been split up by section.

## ***L.2. Description of input and output***

The following parameters are used as input and outputs for the thermal property estimation code and the prompts with their expected values are described below. Users

will typically have to experimentally determine values for the different options if they find that the initial results are not satisfactory (i.e. unusual measurements).

### Input

It should be noted that the code here does require loading a pre-processed file that was output from the pyrometry code (pyroCam) which is detailed in Appendix D. For brevity, the variable descriptions for variables that are loaded in and not later used in the code are not provided. Details on the variables that are automatically loaded can be reviewed in Appendix D.

**Table L-2:** Input options and prompts for pyrometry code with associated descriptions.

Prompt	Description
Pick a processed mat file	<p>Users are asked to navigate to the desired pre-processed .mat file output from the pyrometry code that they would like to be analyzed using the graphical user interface of their machine. The program will automatically save the path that the user has navigated to for future use in saving and organizing files. This will automatically load the variables that are in the pre-processed .mat file.</p> <p>Variables stored: files, path, filename, pathstr, name, ext, raw_image_array, im_height, im_width, num_pixels, num_images</p>
Input image to analyze:	<p>User entered value to determine the image that will be analyzed as part of the thermal property estimation code.</p> <p>Input must be a number.</p> <p>Variables stored: img</p>
Select new line for analysis with GUI? (y/n):	<p>User option to select a new line for thermal property estimation analysis using either a graphical user interface or a number. This line will be the one used to estimate the thermal properties in the horizontal direction.</p>

	<p>Input options are y or n.          If y is selected, the brightest frame of the video will be shown and users will be able to click a line that will be used. Vertical position will be used to estimate the line to do calculations on.          If n is selected, users are asked to enter a value for a line to do calculations on.          If something other than y or n is selected, then the program will execute as if n was chosen.</p> <p>Variables stored:          imgline</p>
Pixel resolution (um/px) :	<p>User entered value for pixel resolution used to record the video that is being processed.</p> <p>Input must be an integer.</p> <p>Variables stored:          px</p>
Smoothing factor (8 default) :	<p>User option to define the smoothing parameter by which to smooth the temperature profile. It had been previously found that the values of the thermal properties are unrealistic if the data was not smoothed.</p> <p>Input must be an integer.</p> <p>Variables stored:          smoothT_factor</p>
Create a single line temperature plot? (y/n) :	<p>User option to create a single line temperature plot with the raw data and smoothed line that had been estimated in the previous section.</p> <p>Input options are y or n.          If y is selected, a figure will be displayed showing the thermal gradient in the sample and the figure will be saved.          If n is selected, the section will be skipped.          If something other than y or n is selected, then the program will execute as if n was chosen.</p> <p>Files stored:          single_line_T_frame&lt;#&gt;_line&lt;#&gt;.fig</p>



Create a time vs. single line temperature plot? (y/n) :	<p>User option to create a figure showing the temperature vs. distance plot for multiple times overlaid. Rarely used, but occasionally useful for general data reviewing.</p> <p>Input options are y or n.  If y is selected, a figure will be displayed showing the thermal gradient in the sample. The figure is not automatically saved.  If n is selected, the section will be skipped.  If something other than y or n is selected, then the program will execute as if n was chosen.</p>
Create a surface plot? (y/n) :	<p>User option to create a 3D surface plot showing the temperature vs. position plot for a single image. Rarely used, but occasionally useful for general data reviewing.</p> <p>Input options are y or n.  If y is selected, a figure will be displayed showing the temperature distribution in the sample. The figure is not automatically saved.  If n is selected, the section will be skipped.  If something other than y or n is selected, then the program will execute as if n was chosen.</p>
Analyze temperature oscillations? (y/n) :	<p>If a user is interested in doing analysis on temperature oscillations, they could run this section of the code, though it is heavily in beta and was never fully developed. For now, it should be skipped. If chosen to be executed, the maximum temperature in each frame will be plotted versus time.</p> <p>Input options are y or n.  If y is selected, a figure will be displayed showing the maximum temperature in the frame vs. time. The figure is not automatically saved.  If n is selected, the section will be skipped.  If something other than y or n is selected, then the program will execute as if n was chosen.</p>
Velocity (cm/s) :	<p>Value for propagation velocity to be used in estimating the thermal properties to change between time and space.</p> <p>Input must be a number.</p> <p>Files generated:  temp.v</p>

Pick window  
region

A graph is shown with the smoothed thermal gradient at the specified frame and line that is being analyzed. Users are to click two points on the graph that define the analysis region on the trailing edge used for fitting the thermal diffusivity.

Users will click two points on the graph that will select the window over which analysis will be performed. After regions are defined, calculations are performed. Calculated values are overlaid on the graph to show users the data that was calculated. Users may rerun this section if the values are not realistic.

Variables stored:  
data.label

### Output

**Table L-3:** Output files for thermal property estimation code with associated descriptions.

Output	Description
single_line_T_frame<#>_line<#>.fig	Figure file that has the temperature vs. position plot for a single frame and line being analyzed.
alpha_analysis_frame<#>_line<#>.fig	Figure file showing thermal diffusivity vs. position overlaid on a smoothed temperature vs. position plot.
thermal_analysis_frame<#>_line<#>.mat	Data file containing all data produced in the calculation related to thermal analysis. Does not include kinetic parameter data.
thermal_analysis_full_frame<#>_line<#>.mat	Data file containing all data produced in the calculation related to thermal analysis saved after the kinetic parameter fitting has been completed.

### **L.3. Variable description**

A full list of variables, their data types, and their descriptions is provided below for reference when making modifications to the code or debugging the program. The variables are listed in alphabetical order.

**Table L-4:** Variable list and description for thermal property estimation code.

Variable	Description
A	Pre-exponential factor determined from nonlinear regression model fitting
alpha.est	Matrix of estimated values for the leading edge thermal diffusivity at each point based on the Boddington equation.
alpha.lead	Averaged value for the leading edge thermal diffusivity.
alpha.T_light_fit	Fit for temperature data for light-emission based temperature. Used as input to ease differentiation for Boddington equation.
alpha.T_light_window	Matrix of temperature data within the temperature region on the leading edge. Temperature is estimated from light emission.
alpha.term1	Term 1 in the Boddington equation for the leading edge evaluated at each point. $(T - T_0)/(t_d - t_r)$ .
alpha.term2	Term 2 in the Boddington equation evaluated at each point. $dT/dt$ .
alpha.term3	Second derivative term for the Boddington equation used in term 3 evaluated at each point. $d^2T/dt^2$ .
ax	Axis handle.
brightframe	Index of the brightest frame in the video.
copypaste	Data to be used for copying and pasting important data into Excel.
d	Handle for patch displayed on the trailing edge of the thermal diffusivity plot.
d_x	Matrix of x-values for box to be displayed highlighting the decay region in the thermal diffusivity plot.
d_y	Matrix of y-values for box to be displayed highlighting the decay region in the thermal diffusivity plot.
data.l	Exponential fit for the left side values.
data.label	Inputs from graphical input by user to determine region of interest on the trailing edge.
data.l_alpha	Calculated values for thermal diffusivity on the left side.
data.l_alpha_mean	Average thermal diffusivity of the left side values.

<code>data.l_alpha_median</code>	Median thermal diffusivity of the left side values.
<code>data.l_d2Tdx2</code>	Calculated values for the second derivative of temperature by position on the left side.
<code>data.l_dTdx</code>	Calculated values for the derivative of temperature by position on the left side using the exponential fit.
<code>data.l_Tx</code>	Calculated values of temperature vs. position on the left side based upon an exponential fit.
<code>data.pos_idx</code>	Index of points nearest to the user-clicked points for the data window.
<code>data.pos_range</code>	Conversion of position to time of the points closest to the user-chosen data window.
<code>data.Tx</code>	Values of temperature vs. position over the window under consideration.
<code>data.v</code>	Steady state propagation velocity.
<code>decay.p1</code>	Point 1 of the decay region based on the fall time function.
<code>decay.p2</code>	Point 2 of the decay region based on the fall time function.
<code>decay.t</code>	Timestamps for the decaying region.
<code>decay.t_total</code>	Total time for the decaying region.
<code>Ea</code>	Estimated activation energy based on the nonlinear regression fit.
<code>Ea_guess</code>	Initial guess for the nonlinear regression fit.
<code>ea.deta_dt</code>	Matrix of values for $d\eta/dt$ based on the Boddington equation and temperatures above the ignition temperature.
<code>ea.eta</code>	Matrix of $\eta$ for each point in the rising edge above the ignition temperature. Trapezoidally integrated from the values obtained in <code>ea.deta_dt</code> .
<code>ea.eta_pass</code>	Matrix of values to be passed into the nonlinear regression fitting.
<code>ea.fit_fn</code>	Fitting function for the temperature as a function of time for the leading-edge temperature above the ignition point. Used for easing differentiation.
<code>ea.fit_model</code>	Exponential model to be used to fit the temperature on the leading edge as a function of time.
<code>ea.gof</code>	Goodness of fit for the leading-edge temperature fit.
<code>ea.p1</code>	Point 1 in the rising edge for the temperature profile above the ignition point.

ea.p2	Point 2 in the rising edge for the temperature profile above the ignition point.
ea.r2	R <sup>2</sup> parameter for the fit of the model for rising edge temperature.
ea.t	Timestamps for the temperature on the leading edge.
ea.t_fix	Fixing timestamps to be normalized so that the first timestamp is 0.
ea.T_light_window	Temperature data for points above the ignition point on the leading edge.
ea.term1	Term 1 in the Boddington equation for the leading edge evaluated at each point over the region above the ignition point. $(T - T_0)/(t_d - t_r)$ .
ea.term2	Term 2 in the Boddington equation evaluated at each point over the region above the ignition point. $dT/dt$ .
ea.term3	Second derivative term for the Boddington equation used in term 3 evaluated at each point over the region above the ignition point. $d^2T/dt^2$ .
ea.upper_lim_l	Upper limit of light emission to be used for the rising edge and determination of data to be used in the activation energy estimation.
ea.upper_lim_T	Temperature estimate for the value stored in ea.upper_lim_l.
EaR	E <sub>a</sub> /R value determined from the nonlinear regression fit of the data determined by the Boddington model.
ext	File extension.
fig	Figure handle.
filename	Filename of the file being analyzed including path.
files	Filename of the file being analyzed.
fitops	Fit options to be used in the nonlinear regression model.
fun	Arrhenius function model to be used for nonlinear regression of data when determining kinetic parameters.
gain	Gain to be used in image display.
grad1	Calculated gradient (K/mm) on the left side.
grad2	Calculated gradient (K/mm) on the right side.

h	Figure handles.
i	Counting variable.
idx	Index of closest point to user input.
idx_max	Index of the maximum value in the smoothed temperature line.
idx_min1	Index of the first point of the smoothed temperature line on the left side.
idx_min2	Index of the last point of the smoothed temperature line on the right side.
img	Image being analyzed.
imgline	Vertical line in image being analyzed.
in1	Inputs of data to be used for nonlinear regression fitting.
keepme.alpha_lead	Leading alpha value to be retained after the program for checking.
keepme.alpha_trail	Trailing alpha value to be retained after the program for checking.
keepme.ea	Activation energy value to be retained after the program for checking.
keepme.ea_err	Error in activation energy value to be retained after the program for checking.
l_chem	Light emission value corresponding to the ignition temperature. It is assumed that no chemistry happens below this temperature.
l_chem_lower	Factor for the lower limit of the light to be used for determining the rise and decay times/windows.
light_black	Black level for temperature based on light emission.
light_corr	Smoothed temperature values for temperature values based on light emission.
light_intensity	Normalized light intensity vs. frame.
light_line	Light intensity over the selected line in the image used for analysis.
name	Name of the file being run.
newlabels	Labels for x-axis of the plot being generated.
num_images	Number of images in the video.
path	Path of the file being analyzed.
pathstr	Path of the new folder to be created.
pos	Position vs. pixel data, units in m.
px	User entered pixel resolution of the video.
r	Handle for patch displayed on the leading edge of the thermal diffusivity plot.

r_x	Matrix of x-values for box to be displayed highlighting the decay region in the thermal diffusivity plot.
r_y	Matrix of y-values for box to be displayed highlighting the decay region in the thermal diffusivity plot.
raw_image_array	Three-dimensional array of pixels, image height x image width x frame.
rFit	Fit function for temperature as a function of light emission.
rFitc	Coefficient values for the light-emission based temperature fit.
rise.p1	Point 1 of the rise region based on the rise time function.
rise.p2	Point 2 of the rise region based on the rise time function.
rise.t	Timestamps for the rising region.
rise.t_total	Total time for the rising region.
rL	Light values to be used for fitting light emission to temperature.
rT	Temperature values (from color ratio pyrometry) to be used for fitting light emission to temperature.
rtable	Combining values of light and temperature.
scaling	Scaling value for plotting data.
setup	Setup data for the video based on the pyrometry code/MatCine.
smoothT_factor	Factor by which the video is smoothed.
smoothT_image	2D smoothed image for the image being processed.
smoothT_line	Single line temperature vs. position data for the frame under consideration using the smoothed data.
soln	Nonlinear regression fitting solution.
t	Timestamps.
T_chem	Temperature used for estimating rise/decay time and kinetic parameter fitting window. Chemistry is assumed to only happen above this temperature.
T_image	Temperature data for each frame imported from pre-processed pyrometry data.
T_light	Light-emission based temperature data.
T_line	Single line temperature vs. position data for the frame under consideration using the raw temperature data.

t2	Temporary variable used while estimating oscillations in maximum temperature.
Tad	Adiabatic flame temperature of the mixture.
tempgray	Gray image conversion variable.
tempimg	Demosaiced image which is converted to grayscale and used as input for light-emission based temperature fit.
test	Used for checking validity of the nonlinear regression model fit.
tframe	Time for each frame.
temp	Temporary variable.
tpx	Time per pixel based on velocity.
v	Velocity (in m/s).
val_max	Maximum value in the smoothed temperature line used for gradient estimation.
val_min1	Minimum value in the smoothed temperature line on the left side used for gradient estimation.
val_min2	Minimum value in the smoothed temperature line on the right side used for gradient estimation.
var0	Matrix of initial guesses for nonlinear regression fitting.
wait	Waitbar handle.
xticks	Tick mark values to be used in plots.

#### ***L.4. Code structure***

The thermal property estimation code is broken up into discrete sections as outlined in the code itself and separated by `%%`. A description of each of these sections and their main functions is below, but the program has been heavily commented for further reference.

**Table L-5:** Code structure for thermal property estimation code.

<b>Section</b>	<b>Description</b>
Data import	User option to import data from the output of a pyrometry estimation run.
Setting analysis section	User input section to define a new frame and line to be used for property estimation.



Data extraction and smoothing	Section with user inputs to choose data smoothing parameter to be used with calculations.
Estimating temperature based on light emission	Section determines a curve fit to correspond light emission with temperature for a given frame in order to extend the dynamic range of temperature measurements in the system. Necessary to estimate rising time, decay time, and leading edge thermal diffusivity.
Determining thermal gradient	Automation of estimating the thermal gradient on leading and trailing edges.
Plot of single line temperature	Section to plot single line temperature vs. position and output of the file.
Plot of multiple single line temperatures	Optional section to plot all of the temperature vs. position plots.
3D temperature plot	Optional section to plot the temperature vs. position as a 3D surface plot for viewing.
Temperature oscillations?	Optional section to estimate the temperature oscillations based upon plotting the maximum temperature in each frame vs. time.
Temperature v. time and alpha analysis - Trailing edge	Calculation section of the code where users are presented with the smoothed temperature data and then given the option to define the window over which to estimate the thermal properties on the trailing edge. Prior to this being run, users must know the propagation velocity.
Temperature v. time and alpha analysis - Leading edge	Calculation section of code where rise time and decay time for the Boddington equation are determined by light-emission based temperature data. The upper limit of the rise temperature should be at the supposed ignition temperature of the mixture. Data for rise and fall time and temperature data is then used to estimate thermal diffusivity assuming that the conversion on the leading edge prior to ignition is 0 ( $\eta = 0$ ). Prior to this being run, users must know the propagation velocity.
Plotting for thermal diffusivity	Section for users to visualize the window for thermal diffusivity and the estimated values.
Save figure and data	Automated saving of thermal analysis figures with thermal diffusivity.

Kinetic parameter fitting setup	Section used to setup terms for kinetic parameter fitting based upon Boddington's equation. Temperature data above the ignition point is used to calculate the first and second derivatives of the temperature profile as well as estimate $d\eta/dt$ using Boddington's equation (alpha, velocity, and rise/decay times were determined earlier in the code).
Boddington Kinetic Parameter Fitting	Section where kinetic parameters are determined using a defined Arrhenius model. Values for a pre-exponential factor (A), reaction order (n), and activation energy are determined using a nonlinear regression model fitting.
Full data package saving	Saves all data including kinetic parameter fits.

### ***L.5. Summary***

This user manual and code description includes the input, output, and structure of the thermal property estimation code developed for gasless reacting materials. If sections of this code are used in other programs or should be cited as part of future research, please reference this dissertation or the corresponding article for Chapter 6.

### ***L.6. MATLAB code***

```

%% Estimating the thermal gradient in a film to determine thermal
diffusivity
% Dylan J. Kline
% Chemical Engineering - Zachariah Research Group
% University of Maryland College Park
% dkline@umd.edu

% UNAUTHORIZED DUPLICATION AND/OR REDISTRIBUTION
% OF THIS SOFTWARE ONLINE OR OFFLINE IS
% STRICTLY PROHIBITED.

clc
clear
close all

% Adds all external functions to working path.
addpath(genpath(fileparts(which(mfilename))))

%% Data import

% Pick .mat file using ui

```

```

[files, path]=uigetfile({'*.mat'}, 'Pick a processed MAT file'); %
GUI selection of files
filename=fullfile(path, char(files)); %
Extracts the filename and path
[pathstr,name,ext]=fileparts(filename); %
separates parts of filename and path

% Extraction of images using pycine (before) or reruns
load(filename) %
should import the data as a matlab file

%% Setting analysis section

% Determine image and line to perform analysis on for the selected
video.
% Ideally, you would pick a central line in a frame where all
img = input('Input image to analyze: '); % image in video to analyze

% Finds the brightest frame and displays the desired image of the
video
% for a user to select a line on which to perform measurements.
for i=1:num_images %
iterate through each frame
% Integrated amount of light hitting the sensor
light_intensity(i)=sum(sum(sum(raw_image_array(:, :, i)))); %
integrate light intensity
end
light_intensity=light_intensity/max(light_intensity); %
normalize the results from intensity
[~,brightframe]=max(light_intensity); %
return the maximum value
gain=4096/max(max(raw_image_array(:, :, brightframe))); %
determine gain for displaying image of vide

figure %
display image with gain
image(double(demosaic(raw_image_array(:, :, img)*gain, 'gbgr'))/4096) %
demosaic and display image
if input('Select new line for analysis with GUI? (y/n): ', 's')== 'y' %
imgline = round(ginput(1)); %
user defined line to select temperature on using GUI
imgline = imgline(2); %
keep only the vertical coordinate
else
imgline = input('Line to analyze in image: '); %
manually entered line in case the user really has a preference
end
close %
close image

px = input('Pixel resolution (um/px): '); %
User input defining the pixel resolution of the recording
tframe = 1/setup.Framerate16; %
Automatically extracted time per frame from video

%% Data extraction and smoothing - Color Image and Temperature Data

T_image = double(Timage);
% temporary variable which extracts the temperatures measured from
the pyrometry code at desired image
T_image(T_image==0)=NaN;
% change 0s to NaNs

```

```

T_line = T_image(imgline,:,img);
% original line data for unsmoothed data at desired line
smoothT_factor = input('Smoothing factor (8 default): ');
% smoothing factor for the 2d smoothing
wait=waitbar(0,sprintf('Make it smooov',num_images));
for i=1:num_images
    smoothT(:, :, i) =
    smooth2a(T_image(:, :, i), smoothT_factor, smoothT_factor); % smoothing 2d
    array for the single image
    waitbar(i/num_images)
end
close(wait)
smoothT_image = smoothT(:, :, img);
smoothT_line = smoothT_image(imgline, :);
% extracting single temperature line

%% Estimating temperature based on light emission
tempimg = raw_image_array(:, :, img); % Pull out
single image for temporary analysis
tempimg(tempimg<64*1.25)=0; % Black level
thresholding.
tempimg = double(demosaic(tempimg, 'gbgr')*gain)/4096; %
Demosaicing image
tempgray=rgb2gray(tempimg); % Gray image
conversion.

light_line = tempgray(imgline,:); % Getting
light intensity for single line in image
light_black = mean(mean(tempgray(1:3, :))); % Getting
black level for area that is presumably at a much lower T.
light_corr = smooth(light_line-light_black, 25); % Smoothing
the light
light_corr(light_corr<0)=0; % Correcting
so that areas below 0 are set to 0.

T_line = smoothT_line'; % Pulling out
the smoothed temperature data for the same line
T_line(T_line==0)=NaN; % Removing 0s
and putting in NaNs.

light_corr = flipud(light_corr); % flipping
corrected light and temperature values to make calculations easy
T_line = flipud(T_line);

% Creating a function to estimate temperature from light
% This was done to improve the sensitivity of our camera below
values that
% would otherwise be impossible for us to see. Essentially, I am
taking a
% string of values and fitting it to a regression line with the form
%  $y=ax^b$ .
% Function form:  $(T-298) = a*(light\_corr)^b$ ;
rL = light_corr; % Corrected light
values
rT = (T_line-298); % Corrected
temperature values
rtable = horzcat(rL, rT); % Making a table
to make life easy
rtable(isnan(rtable(:, 2)), :) = []; % Removing NaN's
for the fit
rFit = fit(rtable(:, 1), rtable(:, 2), 'power1'); % Using a power
function to fit the curve

```

```

T_light=feval(rFit,light_corr)+298; % Back
calculating temperature with the new curve

%% Determining thermal gradient
% An attempt at automating finding the thermal gradient in a system
that
% only has a rising and falling slope. This would probably break
down if
% there were steps in the front.

% left gradient
[val_max, idx_max] = max(smoothT_line); % find
the location of the maximum and the value in the smooth line
temp = ~isnan(smoothT_line); %
temporary variable with the smooth line and the NaNs removed
idx_min1 = find(temp,1,'first'); % find
the first index of the reduced smoothed temperature line
val_min1 = smoothT_line(1,idx_min1); % find
the first value of the reduced smoothed temperature line
grad1 = (val_max-val_min1)/(px/1000*(idx_max-idx_min1)); %
Determine the thermal gradient

% right gradient
idx_min2 = find(temp,1,'last'); % find
the last index in the reduced smoothed T line
val_min2 = smoothT_line(1,idx_min2); % find
the last value in the reduced smoothed T line
grad2 = (val_max-val_min2)/(-px/1000*(idx_max-idx_min2)); %
calculate the thermal gradient

%% Plot of single line temperature
% Simple plot to show the temperature of the desired line in the
image
% There is an option to plot the smoothed temperature line.
if input('Create a single line temperature plot? (y/n): ','s')== 'y'
    fig=figure('Color',[1 1 1]);
    set(fig,'defaultAxesColorOrder',[0 0 0]; [0 0 0]);

    hold on
    box on

    scatter(px/1000*[0:im_width-1],T_line,'k')
    plot(px/1000*[0:im_width-1],smoothT_line,'-r','LineWidth',2);
    xlabel('Distance (mm)')
    ylabel('Temperature (K)')
    legend('Raw Data','Smoothed Data','Location','southeast')
    hold off
    ax=gca;
    ax.FontSize=14;
    title(sprintf('Thermal Gradient\nFrame %d, Line %d',img,imgline))
    hold off

savefig(horzcat(pathstr,sprintf('/single_line_T_frame%d_line%d',img,i
mgline)))
end

%% Plot of multiple single line temperatures
if input('Create a time vs. single line temperature plot? (y/n):
','s')== 'y'
    fig=figure('Color',[1 1 1]);
    set(fig,'defaultAxesColorOrder',[0 0 0]; [0 0 0]);

```

```

hold on
box on

for j=1:50:size(smoothT,3)
    plot(px/1000*[0:im_width-1],smoothT(imgline,:,j));
    xlabel('Distance (mm)')
    ylabel('Temperature (K)')
    ax=gca;
    ax.FontSize=14;
end
hold off
end

%% 3D temperature plot
% A nice little feature that doesnt have a lot of use, but looks
cool.
% Originally used to see and check the uniformity of the gradient
from
% different angles
if input('Create a surface plot? (y/n): ','s')== 'y'
    figure
    %surf(Timage(:,:,img)) % create a surface plot
    surf(smoothT_image);
    colormap(jet) % colormap the surface with the nice
colors
    shading interp % interpolate shading
    title(sprintf('Thermal Gradient\nFrame %d',img))
end

%% Temperature oscillations?
% This section was designed to analyze the auto-oscillations in
% temperature we would see when observing the reaction fronts in the
% camera. It is difficult to perform phase analysis with this data
as of
% now and this section has been put in as a test section.
if input('Analyze temperature oscillations? (y/n): ','s')== 'y'
    figure
    hold on
    t2 = Tmax(isnan(Tmax(:,2))==0,:);
    timeend = 2.5;
    t2(t2(:,1)> 2.5,:)=[];
    t2(:,2) = smooth(t2(:,2),15);
    %scatter(Tmax(:,1),Tmax(:,2),'k')
    plot(t2(:,1),t2(:,2),'r')
    hold off
end

%% Temperature v. time and alpha analysis - Trailing edge
% A rather valiant attempt at measuring the thermal diffusivity of
the
% samples that were recorded. This section begins with a user being
shown
% a graph that allows them to choose a selection window over which
to
% perform analysis. An polynomial curve fit is used to create a
% function that can be used to extract the second derivative of the
% temperature profile with respect to distance.
% Alpha values are then calculated using the fits and
% the fits are used throughout the rest of the code to make other
nice
% graphs that unlock the secrets of SHS.

% This method of analysis is only applicable to regions that are

```

```

% considered to be "remote inert zones" that presumably have no
chemistry
% going on. In the case of the RAIL article, we have chosen to look
at
% temperatures below the observed phase transition.

% This took so much time, many hours of frustration, a
% variety of textbooks, and then just kinda became what it is. I
admit it
% isn't pretty, but it works and I guess that is the engineering
way.

close all

clearvars temp

% Sometimes, when reloading data you would end up with some
variables
% missing
if exist('im_width')==0 % Checking to see if
the image width was retained % Determine image
    im_width = size(smoothT_line,2); % Determine image
width if not
end
if exist('px')==0 % Checking to see if
the pixel resolution was retained % Setting to 20um/px
    px = 20; % Setting to 20um/px
by default for this experiment
end
if exist('v')==1 % Checking to see if
velocity was retained or variable was cleared % Retrieving velocity
    data.v = v; % Retrieving velocity
if it was left in from the save file
else
    data.v = input('Velocity (cm/s): ')/100; % Prompt user to
enter velocity if it was not retained
end

v = data.v; % Retaining velocity
in the event that the user wants to rerun this section

%----- Figure Modification -----%
fig=figure('Color',[1 1 1]);
set(fig,'defaultAxesColorOrder',[0 0 0]; [0 0 0]);
hold on
box on

% Making a graph that shows where data will be analyzed
yyaxis left
h(1) = scatter(px*1e-6*[0:im_width-
1],smoothT_line, '.', 'MarkerFaceColor',[0.5 0.5 0.5]);
pos = px*[0:im_width-1]*1e-6;
xlabel('Distance (m)')
ylabel('Temperature (K)')
ax=gca;
ax.FontSize=14;
title(sprintf('Thermal Gradient\nFrame %d, Line %d\nPick window
region',img,imgline))
set(gca,'ycolor','r')

data.label = ginput(2); % load graph and have
user click two inputs for heating

```

```

for i=1:size(data.label)
    [~,idx] = min(abs(pos-data.label(i,1)));    % find closest point
to user-clicked point
    data.pos_idx(i) = idx;                    % record index of
closest point
    data.pos_range(i) = pos(idx);              % record time of
closest point
    vline(data.pos_range(i));                  % place vertical line
in the window for user to see
end
%----- END Figure Modification -----%

% Establishing a variable for Temperature vs. position using data
entered
% by the user
data.Tx = [pos(data.pos_idx(1):data.pos_idx(2))',
smoothT_line(data.pos_idx(1):data.pos_idx(2))'];
data.Tx(isnan(data.Tx(:,2)), :) = [];

% Cooling side - Fitting
% Using the external polyfix function to set the apex of the curve to
be continuous with a derivative of 0

% Polynomial fit
%fo = fitoptions('poly2','lower',[-Inf -Inf -Inf],'upper',[0 Inf
Inf]);
%data.l = fit(data.Tx(:,1),data.Tx(:,2),'poly2',fo);
data.l = fit(data.Tx(:,1),data.Tx(:,2),'expl');
% Consider switching to expl
data.l_Tx = feval(data.l,data.Tx(:,1));
% Evaluating the polynomial determined by polyfix
[data.l_dTdx, data.l_d2Tdx2] = differentiate(data.l,data.Tx(:,1));
% Taking the first and second derivative using the built in
differentiation functions.

% Exponential fit
% data.l = fit(data.Tx(:,1),data.Tx(:,2),'expl');
% data.l_Tx = feval(data.l,data.Tx(:,1));
% [data.l_dTdx, data.l_d2Tdx2] = differentiate(data.l,data.Tx(:,1));

% Cooling side - Alpha calculations
data.l_alpha = data.v .* data.l_dTdx ./ data.l_d2Tdx2;
% Calculated thermal diffusivity
data.l_alpha_mean = nanmean(data.l_alpha);
% Mean of the thermal diffusivities calculated on the right
data.l_alpha_median = nanmedian(data.l_alpha);
% Median of the thermal diffusivity

%----- Figure Modification -----%
% Plotting thermal diffusivity of cooling side
yyaxis left
h(2) = plot(data.Tx(:,1),data.l_Tx(:,1),'-r','LineWidth',2);

% Plotting thermal diffusivity of cooling side
yyaxis right
h(3) = plot(data.Tx(:,1),data.l_alpha,'-b','LineWidth',2);
ylabel('Thermal diffusivity (m^2/s)')
legend(h([1 2 3]),'Raw Data','Fit
Polynomial','\alpha','Location','northwest');
hold off
%----- END Figure Modification -----%

```



```

xticks = get(gca,'xtick');
scaling = 1000;
newlabels = arrayfun(@(x) sprintf('%i', scaling * x), xticks, 'un',
0);
set(gca,'xticklabel',newlabels)
xlabel('Distance (mm)')

%output = [data.l_alpha_mean std(data.l_alpha) min(data.l_alpha)
max(data.l_alpha) data.r_alpha_mean std(data.r_alpha)
min(data.r_alpha) max(data.r_alpha) data.pos_idx(1) data.pos_idx(2)];
%clipboard('copy',output)

%% Temperature v. time and alpha analysis - Leading edge
% An arguably more valiant effort to estimate the thermal
diffusivity of
% the leading edge of this system. Essentially, what happens is the
% light signal is converted to a generalized intensity plot and then
built
% in functions are used to estimate the rise and fall times of the
% reaction. The light thresholds for "rise" and "fall" time are
based on
% what the expected light intensity would be for the given fit.
Since we
% can assume that no chemistry is happening below this point, we can
set
% the d(eta)/dt of the Boddington equation equal to 0 and since we
know
% temperature, we are able to pull out a thermal diffusivity. We
have
% chosen to use the lowest value for thermal diffusivity since it
would be
% the most appropriate value to choose for estimating activation
energy.
% For reasons that will make sense when looking at the Boddington
% equation, it is necessary to flip the x-axis for the temperature
profile
% to avoid complications with the derivatives.

clearvars alpha

rFitc = coeffvalues(rFit); % Extracting
coefficients for the fit
T_chem = 663+273; % Presumably, no
reaction occurs before the Al melts
%T_chem = 1000;
l_chem = ((T_chem-298)/(rFitc(1)))^(1/rFitc(2)); % Converting the
Al melting temperature into a light signal
%l_chem_post = ((1000-298)/(rFitc(1)))^(1/rFitc(2));

tpx = 20e-6/(v); % Converting position
to time
t = [0:tpx:(size(light_corr,1)-1)*tpx]; % Making a table
with x values and time

clearvars rise decay

l_chem_lower = 0.25; % Percent of the
light to be used as the lower bound. Right now, it is set to 5% of
the light expected for the chemistry threshold

[~,rise.p2,rise.p1] =
risetime(light_corr,'PercentReferenceLevels',[0.1

```

```

99.9], 'Tolerance', 0.01, 'StateLevels', [l_chem_lower*l_chem l_chem]); %
Estimating the rise and fall times/limits
rise.p1 = round(rise.p1); % Point 1 for the
rise time
rise.p2 = round(rise.p2); % Point 2 for the
rise time
rise.t = t(rise.p2:rise.p1)'; % Extracting the
timestamps for the leading edge
rise.t_total = abs(rise.t(1)-rise.t(end)); % Extracting the
total time of the leading edge

[~, decay.p2, decay.p1] =
falltime(light_corr, 'PercentReferenceLevels', [0.1
99.9], 'Tolerance', 0.01, 'StateLevels', [l_chem_lower*l_chem l_chem]);
decay.p1 = round(decay.p1); % Point 1 for the
decay time
decay.p2 = round(decay.p2); % Point 2 for the
decay time
decay.t = t(decay.p1:decay.p2)'; % Extracting the
timestamps for the trailing edge
decay.t_total = abs(decay.t(1)-decay.t(end)); % Extracting the
total time for the trailing edge

% Fitting Equation to Smoothed Calc Data
clearvars alpha
%fo = fitoptions('poly2', 'lower', [0 -Inf -Inf], 'upper', [Inf Inf
Inf]);
T_light = feval(rFit, light_corr) + 298; % Back
calculating temperature with the new curve
T_light = smooth(T_light, 5); %
Smoothing the calculated temperature plot
alpha.T_light_window = T_light(rise.p2:rise.p1); %
Window of temperatures to be used for the leading edge alpha analysis
alpha.T_light_fit = fit(rise.t, alpha.T_light_window, 'exp1'); %
Fitting the leading values of alpha to a polynomial curve

alpha.term1 = (feval(alpha.T_light_fit, rise.t) -
298) / abs(decay.t_total - rise.t_total); % First term in the Boddington
Eqn (T-T0/td-tr)
[alpha.term2, alpha.term3] = differentiate(alpha.T_light_fit, rise.t);
% Second and third terms (Derivatives only) for the Boddington Eqn
(dT/dt and d2T/dt2)

alpha.est = (alpha.term1 + alpha.term2) .* (v^2) ./ (alpha.term3); %
Estimated values for the alpha assuming that the deta/dt==0, m2/s
alpha.lead = mean(alpha.est); %
Minimum value to be used for alpha.

%% Plotting for thermal diffusivity
%----- Figure Modification -----%
fig = figure('Color', [1 1 1]);
set(fig, 'defaultAxesColorOrder', [[0 0 0]; [0 0 0]]);
hold on
box on

% Making a graph that shows where data will be analyzed
%yyaxis left
temp = flipud(T_light);
temp(~isnan(smoothT_line)) = NaN;
h(1) = scatter(px*1e-3*[0:im_width-
1], temp, '*', 'MarkerEdgeColor', [0.65 0.65 0.65]);
h(2) = scatter(px*1e-3*[0:im_width-1], smoothT_line, '.k');
xlabel('Distance (mm)')

```

```

ylabel('Temperature (K)')
ylim([300 1800]);
xlim([0 20]);
ax=gca;
ax.FontSize=14;

% Plotting thermal diffusivity of cooling side
h(3) = plot(data.Tx(:,1)*1000,data.l_Tx(:,1),'-r','LineWidth',3);

%yyaxis right
%h(3) = plot(px*1e-3*[0:im_width-1],flipud(light_corr),'-
.b','LineWidth',2);
%set(gca,'ycolor','b')
ylabel('Light Intensity (arb. units)')

r_x = [pos(size(pos,2)-rise.p1)*1000 pos(size(pos,2)-rise.p2)*1000
pos(size(pos,2)-rise.p2)*1000 pos(size(pos,2)-rise.p1)*1000];
r_y = [298 298 2000 2000];
r = patch(r_x,r_y,'green','EdgeColor','none','FaceAlpha',0.3);
r.FaceColor = '#77AC30';

d_x = [pos(size(pos,2)-decay.p1)*1000 pos(size(pos,2)-decay.p2)*1000
pos(size(pos,2)-decay.p2)*1000 pos(size(pos,2)-decay.p1)*1000];
d_y = [298 298 2000 2000];
d = patch(d_x,d_y,'blue','EdgeColor','none','FaceAlpha',0.2);
d.FaceColor = '#0072BD';
legend('Temperature (light int.)','Temperature (color
ratio)','Fit','location','northwest');

%----- END Figure Modification -----%
%temp = pos(1024-rise.p1:1024-rise.p2)*1000;
%temp = pos(1024-decay.p2:1024-decay.p1)*1000;

%% Save figure and data
if exist('pathstr') == 0
    pathstr=uigetdir('Pick corresponding folder'); % GUI selection
of files
end
savefig(horzcat(pathstr,sprintf('/alpha_analysis_frame%d_line%d',img,
imgline)))
save(horzcat(pathstr,sprintf('/thermal_analysis_frame%d_line%d_v7',im
g,imgline)), 'img', 'imgline', 'smoothT_factor', 'smoothT_image', 'smoothT
_line', 'px', 'im_width', 'v');

%% Kinetic parameter fitting setup
% To do this, we are going to be using the Boddington equation to
estimate
% the value for deta/dt. The alpha value to be used is the lowest
value in
% the calculated values from the leading edge since this would be
the
% remote inert region. The rise and decay times were calculated from
% earlier sections in the code related to finding the leading edge
alpha
% value. tau_ad is Tad-298. Using the risetime function, we identify
% regions where the temperature is above the chemistry initiation
% temperature and fit that temperature data to a curve (hyperbolic
tangent
% was used since it would have a plateau at the peak and eliminated
% inconsistencies in data). These temperature values are then
% resubstituted into the Boddington equation to estimate the deta/dt
at

```

```

% each time in the rising curve. Numerical trapezoidal integration
can be
% used to estimate the total conversion of the reaction and then can
be
% sent to an Arrhenius plot to estimate activation energy.

clearvars ea

Tad = 3765.0; % Isobaric Adiabatic flame temperature for Zr C
stoichiometric reaction from CHEETAH

ea.upper_lim_l = 0.9*max(light_corr); % Upper limit for the
light to be used for finding the window of interest
ea.upper_lim_T = rFit(ea.upper_lim_l)+298; % Calculating the
temperature that the "upper light limit" corresponds to (for
checking)
[~,ea.p2,ea.p1] = risetime(light_corr,'PercentReferenceLevels',[0.1
99.9],'Tolerance',0.01,'StateLevels',[1_chem ea.upper_lim_l]); %
Estimating the rise and fall times/limits
ea.p1 = round(ea.p1); % Point 1 of the rise
time for the Ea window of interest
ea.p2 = round(ea.p2); % Point 2 of the rise
time for the Ea window of interest
ea.T_light_window = smooth(rFit(light_corr(ea.p2:ea.p1))+298,5); %
Calculating the temperature based on light emission for the Ea region
of interest
ea.t = t(ea.p2:ea.p1)'; % timestamps for the
Ea region of interest
ea.t_fix = ea.t-ea.t(1); % Fixing timestamps
for the region of interest since it makes fitting much better

% Model fit
% A hyperbolic tangent with an exponential term was used as the
model
% since it has a plateau and does not drop below 0. In principle,
% conversion of an irreversible reaction should follow a very
similar
% trend.
ea.fit_model = fittype( @(a,b,c,x) a*tanh(b*x)+c); % Custom model
for fitting
[ea.fit_fn,ea.gof] = fit(ea.t_fix,ea.T_light_window,ea.fit_model); %
Getting fitting parameters and storing goodness of fit
ea.r2 = ea.gof.rsquare; %
Storing r2 in case anybody cares

ea.term1 = (feval(ea.fit_fn,ea.t_fix)-298)/abs(decay.t_total-
rise.t_total); % Evaluating the first term in the Boddington equation
over the region of interest
[ea.term2,ea.term3] = differentiate(ea.fit_fn,ea.t_fix);
% Estimating the second and third terms (derivative only) of the
Boddington equation
ea.deta_dt = (ea.term1+ea.term2-(alpha.lead/(v^2)*ea.term3))/(Tad-
298); % Boddington equation, calculate deta/dt
ea.eta = cumtrapz(ea.t_fix,ea.deta_dt);
% Estimate the actual conversion based on cumulative trapezoidal
integration

ea.eta_pass = [ea.t, ea.T_light_window, ea.deta_dt, ea.eta];
% Data to pass back to the main data variable

```

```

%ea.eta_pass = ea.eta_pass(ea.eta_pass(:,4)>0 &
ea.eta_pass(:,4)<1,:); % Reducing data down to low conversion
values... not used.
% Plots if you want them.
%plot(1./ea.eta_pass(:,2),log(ea.eta_pass(:,3)))

%% Boddington kinetic parameter fitting
clearvars cospaste soln
% Clearing variable for copying and pasting into excel for checking
cospaste = [ea.t_fix, ea.deta_dt, ea.eta, ea.T_light_window];
% Variable to create for copying and pasting into excel

clearvars in1 A n EaR Ea soln
% Clearing solution variables
in1 = [ea.t_fix, ea.eta, ea.T_light_window, ea.deta_dt];
% temporary variable with inputs necessary to solve the nonlinear
regression
in1 = in1(in1(:,2)>0 & in1(:,2)<1,:);
fun = @(var,in1) var(1).*(1-in1(:,2)).^var(2).*exp(-
var(3)./in1(:,3)); % Model for the nonlinear regression where "var"
is the matrix of coefficient variables
Ea_guess = 65;
% Starting initial guess for Ea
var0 = [100, 3, Ea_guess/8.3145*1000];
% Initial guess estimations for the nonlinear regress [A,n,Ea/R]
fitops = statset('fitnlm');
fitops.MaxIter = 1000;
soln = fitnlm(in1,in1(:,4),fun,var0,'Options',fitops);
% Nonlinear regression solution

A = soln.Coefficients{1,1};
% Extracting value for A coefficient
n = soln.Coefficients{2,1};
% Extracting value for n coefficient
EaR = soln.Coefficients{3,1};
% Extracting value for Ea/R coefficient
Ea = EaR*8.3145/1000
% Calculating Ea
cospaste_check = [A,n,Ea];
% In case someone wants to verify the model in excel

test = A.*(1-in1(:,2)).^n.*exp(-EaR./in1(:,3));
% Calculating values for checking the nonlinear regression fit

%----- Figure Modification -----%
figure('Color',[1 1 1]);
set(gcf,'defaultAxesColorOrder',[0 0 0]; [0 0 0]);
hold on
box on
scatter(1./in1(:,3),log(in1(:,4)),55,'w','filled','MarkerEdgeColor',[
0 0 0],'LineWidth',1);
plot(1./in1(:,3),log(test),'b','LineWidth',2);
legend('Calculated','Fit');
xlabel('T^{-1}')
ylabel('$\ln(\frac{d\eta}{dt})$', 'Interpreter','latex')
title(sprintf('E_a = %3.1f kJ/mol',Ea));
ax=gca;
ax.FontSize=14;
hold off
%----- END Figure Modification -----%

%% deta/dt vs. time
close all

```

```

%----- Figure Modification -----%
figure('Color',[1 1 1]);
set(gcf,'defaultAxesColorOrder',[[0 0 0]; [0 0 0]]);
hold on
box on
scatter(in1(:,1),in1(:,4),55,'w','filled','MarkerEdgeColor',[0 0
0],'LineWidth',1);
plot(in1(:,1),test,'b','LineWidth',2);
legend('Measured (Boddington Eq.)','Fit','location','northeast');
xlabel('time (s)')
ylabel('$d\{\eta\}/dt$','Interpreter','latex')
%title(sprintf('E_a = %3.1f kJ/mol',Ea));
ylim([0 5])
xlim([0 0.2])
yticks([0:1:5])
ax=gca;
ax.FontSize=14;
hold off
%----- END Figure Modification -----%

%% Full data package saving
save(horzcat(pathstr,sprintf('/thermal_analysis_full_frame%d_line%d_v
7',img,imgline)));
keepme.ea = data.ea;
keepme.ea_err = data.ea_err;
keepme.alpha_lead = alpha.est;
keepme.alpha_trail = data.l_alpha;

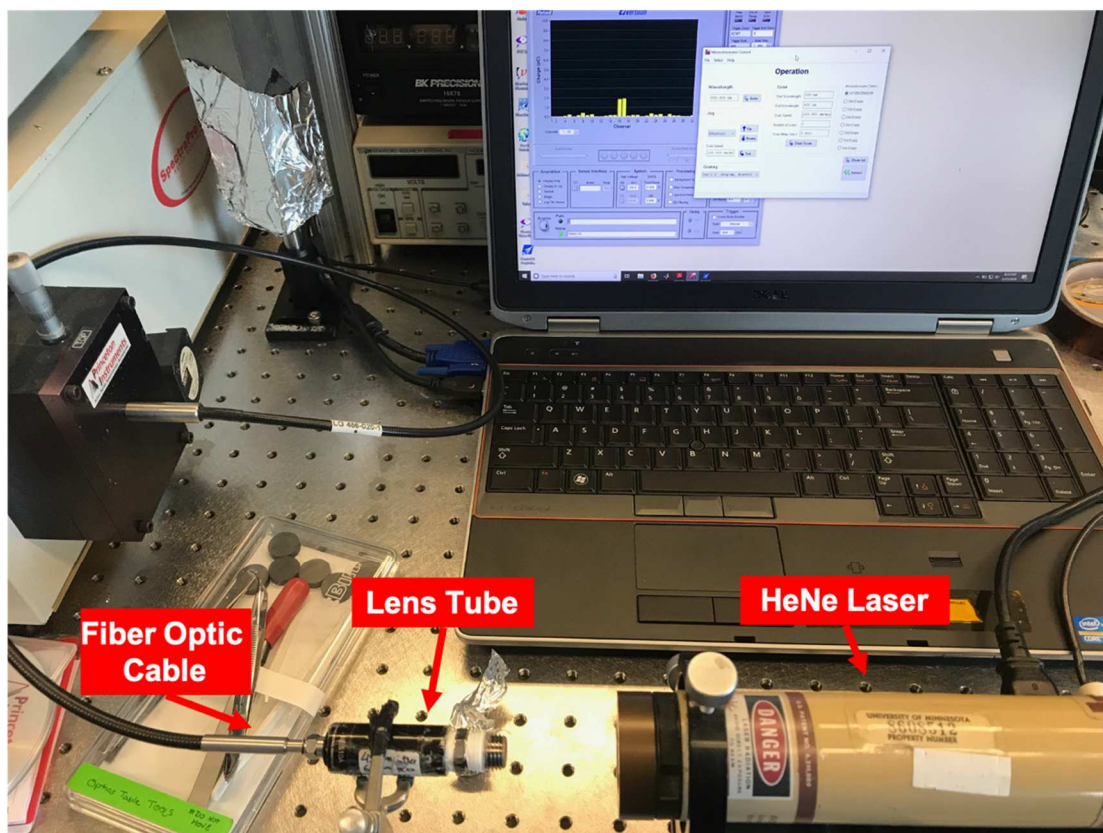
```

## Appendix M: Calibrating a spectrometer and 32-channel photomultiplier tube for temperature measurements

### *M.1. Procedure for calibrating camera pyrometer*

Before running experiments in the constant volume combustion cell (AKA pressure cell) for temperature measurements using the high-speed spectrometer, it is important to calibrate the spectrometer to determine the channel gains on the 32-channel multichannel photomultiplier tube (MC-PMT). It should be noted here that the spectrometer and MC-PMT gains will need to be recalibrated in the event of a power cutoff to the spectrometer. Typical voltage ranges for calibration on the MC-PMT are -500 - -600 V in 25 V increments. A procedure is described below.

1. Retrieve HeNe laser (or light source with a single known wavelength), safety glasses for the laser, and the 1" OD diameter lens tube attachment for the pressure cell.
2. Attach the lens tube to the fiber optic cable entering the spectrometer and turn on computer attached to spectrometer. Set up HeNe laser to shine into the lens tube. (see Figure M-1).

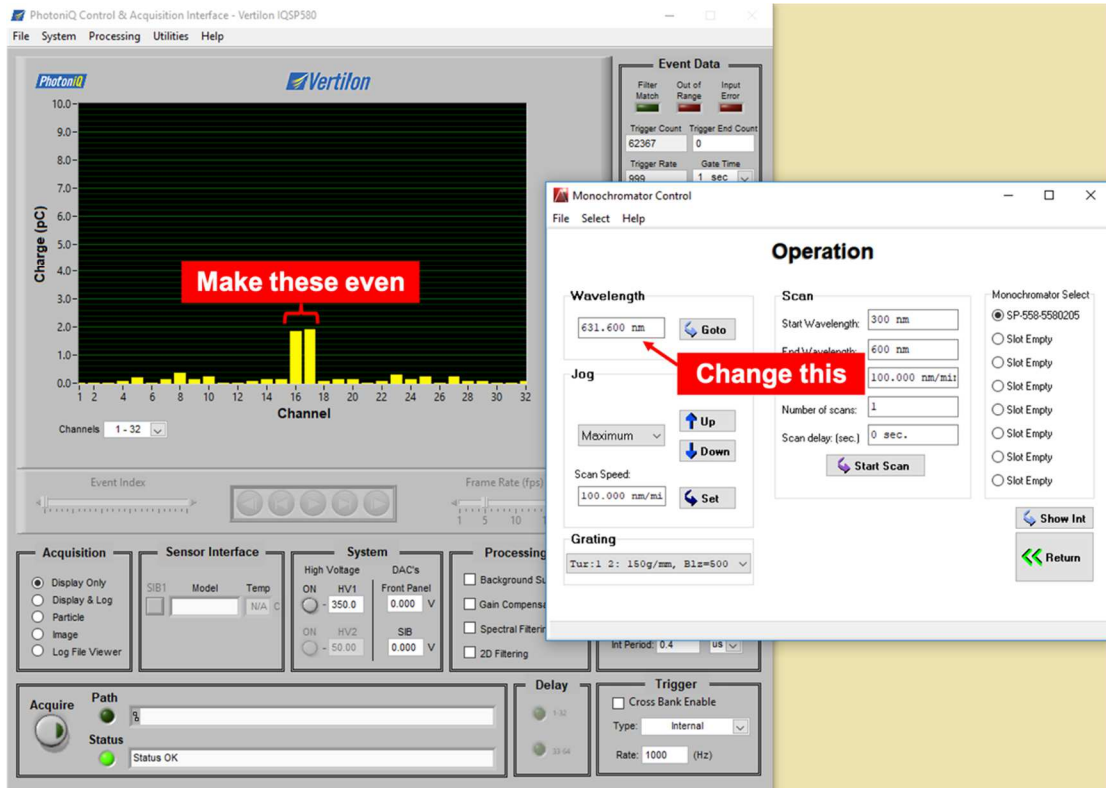


**Figure M-1:** Example setup of the HeNe laser shining into the lens tube for the pressure cell and fiber optic cable leading into the spectrometer.

3. Power on Vertilon data acquisition unit (DAQ) connected to the 32-channel MC-PMT. Enable high voltage supplies under System > High Voltage Supplies and turn on the high voltage supply to -350.0 V on HV1. Set the trigger to “Internal.” Set the “Acquisition” box in the GUI to “Display Only” and click “Acquire.” Data should begin reading in on the Vertilon software.
4. Open up the Acton MonoChromator Control software on the computer. Make sure that the turret with the 150g/mm grating is selected.
5. Power on laser and monitor signal on Vertilon software to make sure that the laser is entering lens tube and is stable.

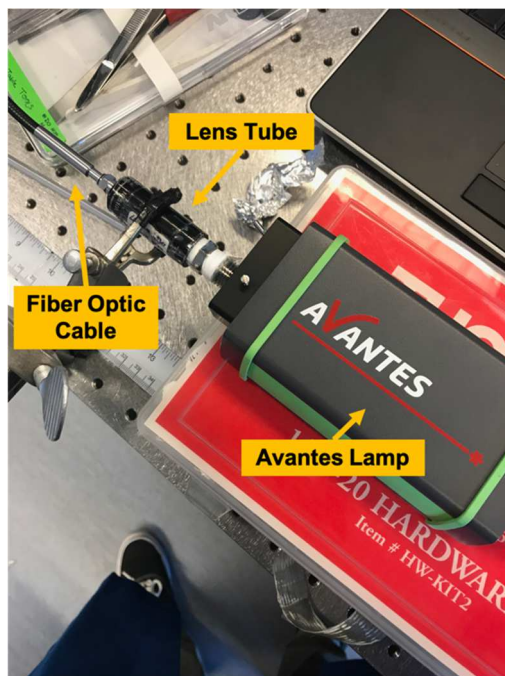


- To get the right settings for the monochromator, users must change the value for “Wavelength” in the Avantes until there is an equal reading between channels 16 and 17 on the Vertilon DAQ software (see Figure M-2).



**Figure M-2:** Screenshot of computer demonstrating desired levels between channels 16 and 17 for spectrometer calibration on the Vertilon software and example screenshot of the Acton MonoChromator Control software.

- Once completed, stop data acquisition on the Vertilon software, turn off the high voltage power supply, power off the laser, and pack it away.
- Retrieve the Avantes pencil lamp and power it on. Allow to equilibrate for 5-10 minutes.
- Place the inlet to the lens tube flush against the outlet of the Avantes lamp (Figure M-3).



**Figure M-3:** Example setup of the Avantes lamp shining into the lens tube for the pressure cell and fiber optic cable leading into the spectrometer for MC-PMT gain calibration.

10. Set the desired voltage on HV1, trigger rate, integration period, and trigger end count on the Vertilon software (refer to the table below for suggested settings). Take a background subtraction at the current settings by covering the inlet to the optical tube. After a successful background subtraction, check the box next to “Background Subtract” under in the “Processing” box. Set the “Acquisition” to “Image,” click the “Acquire” button and choose a place/name for the file to be saved.
11. Repeat step 10 for the range of voltages to be used during experiments. In the past, this has been done from -500 - -600 V in 25 V increments. The maximum voltage that should be applied to the MC-PMT is -650 V. Before starting the next run, check the recorded log file using the “Log File Viewer” under the “Acquisition” box. Also, be sure that before taking another background

subtraction that the voltage is set to the voltage of the next increment, the check box next to background subtract is disabled, the desired integration time is set, and that the trigger end count is set to 0.

**Table M-1:** Recommended settings for voltage, trigger rate, integration period, and trigger end count.

<b>Voltage (HV1)</b>	<b>Trig. Rate</b>	<b>Int. Period</b>	<b>Trig. Count</b>
-500 V	1000 Hz	5 ms	150
-525 V	1000 Hz	5 ms	150
-550 V	1000 Hz	5 ms	150
-575 V	1000 Hz	5 ms	150
-600 V	1000 Hz	5 ms	150

12. Once finished, convert the .log files to .txt using the “Log File Converter” under the “Utilities” menu in the Vertilon software. Place all of the converted .txt files in a single folder for efficient processing using the channel gain calibration code in the next section.

### ***M.2. Program for generating calibration file for 32-channel MC-PMT***

A program has been written to generate a calibration file that is read into the spectrometer temperature measurement program. After generating the spectrometer files and converting the files to a .txt format, place all of the .txt files in a single folder and run the code below. The `PCellCorrFactorND2` variable in the workspace generated after the program runs should be saved in a new .mat file, named with the date of the calibration, and the main temperature measurement code should have the filename/path for the calibration updated in the “Filtering Data – Spectrometer” section to point to the new file.

Running this program has dependencies on other programs that are in the Zachariah Group Drive at Volumes/users/Kline/Codes/Pressure-Cell-Pyrometry/. Notably, channelWavelengthGen.m, lamp\_avantes.mat, importmat.m, importpar.m, and importvolt.m. The opening lines of the code should automatically add the files to the working path if the group drive is loaded as the Z:\ drive on a PC.

```
%% Spectrometer Gain Calibrations
% Dylan J. Kline
% Chemical Engineering - Zachariah Group
% University of Maryland College Park

% This code is designed to create a spectrometer calibration file
for use
% with the high speed 32 channel spectrometer in the Zachariah
Group.
% Before starting, place all of the converted .txt spectrometer
files in a
% single folder and then the code will automatically perform the
% calculations on the files placed in that folder. For detailed
procedures
% on calibrating the spectrometer, please refer to the

%% Data & Script Importing

% Adds all external functions to working path.
addpath(genpath(fileparts(which(mfilename))))

% Imports the data for the avantes lamp that we use to calibrate
with
chWave = channelWavelengthGen(663,150);
% HgNe pencil lamp has a wavelength of 663 nm and we center it
between channels 16/17
load('/Volumes/users/Kline/Pressure Cell Codes/Calibration
Data/lamp_avantes.mat'); % loads in matlab data for lamp
lamp_avantes(:,1)=lamp_avantes(:,1)*1e-9;
% changes the data to units of m from nm
f=fit(lamp_avantes(:,1),lamp_avantes(:,2),'poly8');
% fits lamp data for interpolation to channel wavelengths
calib_spectra = [chWave(3,:) ', f(chWave(3,:))];
% calculates intensities at different wavelengths
calib_spectra(:,3) = calib_spectra(:,2)./max(calib_spectra(:,2));
% creates normalized value for the intensities

%% File Import and Looping

% Get data file folder
pathname=uigetdir('Select folder with calibration data');

% creates a file list for all of the calibration files
filelist = dir(fullfile(pathname,'*.txt')); % creates a file list
matching the .txt
```

```

num_files = size(filelist,1);           % determines the
number of files

% will loop through the files and perform the calculations necessary
and
% output a variable that can be used as the MC-PMT gains.
for i=1:num_files
    file = fullfile(filelist(i).folder,filelist(i).name); % picks
    file and name

        spectral_data = importmat(file);           % imports
the data from the file
        spectral_data(spectral_data<0)=0;           %
eliminates weird negative values (which shouldnt exist)
        spectra_avg = mean(spectral_data(:,6:37),1)'; %
averages the data over the entire collection period
        spectra_avg_norm = spectra_avg./max(spectra_avg);

        par = importpar(file);                     % not
really sure what this does
        t_int = par(1,1);                           %
extracts integration time in us
        srate = par(3,2);                           %
extracts the spectrometer rate in Hz
        volt = importvolt(file);                     %
extracts the voltage used on the PMT

        PCellCorrFactorND2(i,:)=[volt, t_int,
(calib_spectra(:,3)./spectra_avg_norm)'];
end

```

## Appendix N: High speed, 32-channel spectrometer temperature measurements in constant volume combustion cell

This appendix will serve as the user manual for the code designed to measure temperatures for coupled pyrometry with the experiments performed in constant volume combustion cell (AKA pressure cell). The code here was rewritten from original versions written for research in Chapter 2 conducted with Dr. Rohit Jacob, but the theory behind the code is the same through each of the versions. Throughout the manual, text in `this font` will represent variables or expressions that are used in a MATLAB command.

### ***N.1. Running pressure cell temperature measurement code***

The executable MATLAB script can be found in the Zachariah Group Drive as Volumes/users/Kline/Codes/Pressure-Cell-Pyrometry/kline\_pressure\_cell\_20200317.m. An online version has also been stored with version control and is available by reasonable request. The code has numerous dependencies that were written by Vertilon (the data acquisition unit manufacturer) and Dr. Rohit Jacob during his time in the Zachariah Group. As an alternative to the Vertilon dependencies for converting .log files, one could convert the .log files using the house built MatVertilon program which is available in the Zachariah Group drive or by reasonable request (this software is much faster at converting large files with >20,000 data points). All of the code dependencies are in the group drive under Volumes/users/Kline/Codes/Pressure-Cell-Pyrometry/. The external functions are well documented and are left for the user to debug when necessary.

**Table N-1:** Dependencies for pressure cell temperature measurement code.

Name	Description
channelWavelengthGen.m	Program designed to take in a wavelength of the spectrometer at channel 17 and the grating with which the spectrometer was calibrated and return an array with the channel wavelengths (chWavelength) and channel widths (chWidth) in the workspace.
importfile.m	Program written by Vertilon to import .txt files generated by the Log File Converter into MATLAB for later processing. Returns all of the signal data for the 32 channel MC-PMT in columns 6-37.
importpar.m	Program written by Vertilon to import parameter data from .txt files generated by the Log File Converter into MATLAB for processing. Specifically used to retrieve the integration time and sample rate of the spectrometer for the experiment.
importvolt.m	Program written by Vertilon to import the voltage applied to the MC-PMT during the experiment, used later to determine which set of gains to use in the temperature measurement calculation.
TempFit.m	Program written by Dr. Rohit Jacob to fit temperature using either a gray body, nonlinear fit (check ~= 1) or transposed fit (check = 1).

## ***N.2. Description of input and output***

The following parameters are used as input and outputs for the pressure cell temperature measurement code and their prompts with expected values are described below. Users may need to run the code a few times to get good results, but this will generally work on the first try. Prior to running this code, users should place their channel data collected from the oscilloscope (C1, C2, and C4 in .dat form) and the corresponding .txt file (generated by Log File Converter in the Vertilon software) or

.mat file (generated by a custom MatVertilon routine) in a single folder with the name of the sample.

### Input

**Table N-2:** Input options and prompts for pressure cell temperature measurement code with associated descriptions.

Prompt	Description
Select folder to open	<p>A graphical user interface prompt that asks users to select a folder which contains the channel data from the oscilloscope (in .dat form) and the spectrometer .txt log file.</p> <p>Variables stored: FolderName, OscFile, SpecFile</p>
Are there any sharp peaks/broken points that are prominent in the spectra? Enter the X-axis positions as an array:	<p>User option to selectively exclude channels in the spectra for temperature fitting. Generally, this is used when there is a species in the tested formulation that would have a very bright molecular emission (i.e. Sodium) or have strong absorption properties (i.e. Iodine).</p> <p>Input must be an array of integers. For physically mixed samples, it is recommended to use [1 5 18 22 23] in addition to any other channels that are not smooth.</p> <p>Variables stored: spec.molemission</p>
Enter 3 channels as an array to test smoothing:	<p>This will generate a figure with the three channels that will be used to check the smoothing that will be applied to the spectrometer channel data for temperature measurement. It is best to use channels that were not excluded in the previous prompt for being too sharp or broken.</p> <p>Input must be an array of integers.</p> <p>Variables stored: testCh</p>
Improve smoothing? (0/1):	<p>User dialog to smooth the spectrometer channel data for temperature measurement. This loop will execute until the user enters 1, giving users multiple opportunities to improve their smoothing.</p>



	<p>Input options are 0 or 1.  Users should enter 1 at least once.  If 0 is selected, the program will stop execution of data smoothing of the spectrometer data.  If 1 is selected, the program will enter the smoothing routine for the spectrometer channel data.</p> <p>Variables stored:  check</p>
Enter smoothing parameter (1-50) :	<p>User dialog to enter the smoothing amount to be applied using the built-in data smoothing toolbox in MATLAB. After smoothing the data, it will be plotted on the figure and the user can assess whether they would like to further smooth the data or accept the data. When smoothing data, make sure that data is not inadvertently being flattened or peaks are being removed.</p> <p>Input must be an integer.</p> <p>Variables stored:  Smoother, CD, spec.channeldata</p>
Is baseline ok? (0/1) :	<p>User option to change the baseline for data filtering. If the baseline appears to be suitably above the noise window in the figure generated to show the spectrometer data, then the baseline is acceptable. Otherwise, users will be asked to enter a new baseline.</p> <p>Input options are 0 or 1.  If 0 is selected, the program will ask users to pick a new baseline.  If 1 is selected, the program will accept the baseline and continue executing.</p> <p>Variables stored:  check</p>
New baseline:	<p>User option to change the baseline used for data filtering in the spectrometer data. If the data being shown in the figure does not seem to be well below the baseline, then the baseline is acceptable. Data below the baseline will be excluded.</p> <p>Input must be a number.</p>

Variables stored:
Baseline

*Output*

At the time of writing this document, FIGURES GENERATED ARE NOT SAVED at the end of this file, but the data used to generate the figures is saved.

**Table N-3:** Output files for pressure cell temperature measurement code and associated descriptions.

Output	Description
<filename>.mat	MATLAB output file containing all of the variables generated when running the code including the pressure, diode, signaling, spectrometer, and temperature data.

**N.3. Variable description**

A full list of variables, their data types, and their descriptions is provided below for reference when making modifications to the code or debugging the program. Unlike many of the other codes in this dissertation, this program takes advantages of the struct architecture to better organize data. The variables are listed in alphabetical order by struct.

**Table N-4:** Variable list and description for the pressure cell temperature measurement code.

Variable	Description
AbsIntensityData	Absolute integrated intensity calculated by using unprocessed channel data, multiplying by the integration time for the calibration, dividing by the spectrometer integration time used by the experiment, and correcting for channel width.
ax1	Axis handle.
ax2	Axis handle.
ax3	Axis handle.

ax4	Axis handle.
Baseline	Baseline value used for filtering data below certain value in spectrometer channel data.
Buffer	Buffer time to ensure that all data relevant is imported for pressure and diode data.
C	Temporary variable used to improve visuals in checking for broken/sharp peaks.
CalibIntTime	Integration time used for the calibration with the associated voltage applied on the MC-PMT.
CD	Temporary data holder for channel data used for smoothing of the channel data from the spectrometer.
ChargeLimit	Charge limit for the MC-PMT based on the current linearity limit. Calculated by multiplying the spectrometer integration time by the current linearity limit (25 $\mu$ A) of the MC-PMT. Points in the spectrometer data above the charge limit will be excluded from temperature calculations.
check	Binary variable used to check whether or not a user is ready to proceed to the next section of code or would like to review the smoothing/baseline parameters.
chWave	A 3x32 maxtrix of the upper (row 1), lower (row 2), and middle (row 3) channel wavelength for the MC-PMT with units in m. Only the middle wavelength is used for the temperature calculations.
chWid	Channel width for the 32 channel MC-PMT in m.
CurrentLimit	Current linearity limit (in $\mu$ A) for the MC-PMT. At the time of this dissertation, the current linearity limit is 25 $\mu$ A. Points in the spectrometer data above the charge limit (calculated using the current limit) will be excluded from temperature calculations.
D	Struct with information collected from the photodiode.
D.data	Diode data recorded from the oscilloscope (C2) with the format [time, smoothed diode channel, 50 Ohm termination conversion to mA].
D.Import	Variable only using data beyond the import start point in the diode signal.

D.ImportStart	Starting point for reduced diode signal data considering the lag time.
D.intLag	Integrated lag time for diode used for data showing.
D.StartLoc	Starting location for the data to be included considering where the user has ended the noise window.
D.Threshold	User entered threshold which denotes where the end of the noise window for the diode signal.
DataFinal.D	Final data to show for the diode signal with the corrected time and smoothed data.
DataFinal.P	Final data to show for the pressure signal with the corrected time and smoothed data.
DataFinal.T	Final data to show for the temperature data with corrected time.
DaqTimeStart	Temporary data variable with the first time of the data acquisition unit. Used for time shifting.
f1	Figure handle.
fig	Figure handle.
FolderName	Folder name used for navigating to directories and saving data.
Gains	Gains for the 32 channel MC-PMT at the voltage corresponding to that used in the experiment. 1x32 array.
h1	Axis handle.
h2	Axis handle.
idx_diode_max	Index of the maximum diode signal recorded in the oscilloscope. Used for timecode matching with the spectrometer.
idx_spec_max	Index of the maximum signal recorded in the spectrometer at the channel of peak sensitivity for the photodiode. Used for timecode matching with the diode.
IntegratedIntensity	Sum of the calculated radiances as a function of time.
j	Counter variable.
LimCol	Column index of data that is above the baseline and below the charge limit.
LimRow	Row index of the data that is above the baseline and below the charge limit.
linestart	Linenumber to be used as the starting point for the spectrometer.

name	Name of the file to be saved based on the name of the folder (all spaces are replaced with underscores).
NaNcounter	Temporary variable used to determine the number of NaNs in a given set of spectrometer data at each timestamp. If there are a significant amount of NaNs at a given time, the data is not used to measure temperature.
normCh	Index of the first location where data is not a NaN in the spectrometer data at a given time.
NumChan	Number of channels on the spectrometer.
OscFile	Oscilloscope file handle to be used for data importing.
Output	Temporary variable with the calculated temperature generated by TempFit.m with the given input information.
P	Struct with information recovered from the pressure sensor.
P.data	Pressure data recorded from the oscilloscope (C1) with the format [time, smoothed pressure data in mV, pressure data in PSI, pressure data in kPa].
P.Import	Variable only using data beyond the import start point in the pressure signal.
P.ImportStart	Starting point for reduced pressure signal data considering the lag time.
P.intLag	Integrated lag time for pressure used for data showing.
P.StartLoc	Starting location for the data to be included considering where the user has ended the noise window.
P.Threshold	User entered threshold which denotes where the end of the noise window for the pressure signal.
preTriggerSamples	Number of pretrigger samples stored in the spectrometer. Used for adjusting the first line of data.
ProcRow	Number of rows to be used in variable initialization for temperature measurement. Size of the number of timestamps that have data valid for temperature measurement.
r	Number of timestamps recorded by the spectrometer.

Radiance	Radiance of the spectrometer data with filtering for data that is below 0.
Smoother	User entered smoothing parameter applied to the spectrometer data.
spec.alldata	All spectrometer data imported using <code>importfile.m</code> . Data is later broken down by type.
spec.channeldata	32 column array with spectrometer charge data as recorded from the spectrometer. Each row represents a different timestamp.
spec.ChIR	IR channel data for the spectrometer data which is smoothed and filtered.
spec.ChIR_NoiseIndex	Noise index for the IR channel in the spectrometer.
spec.ChIR_Threshold	Threshold for background noise in the IR channel of the spectrometer.
spec.ChIR_Start	Starting point for IR channel data in the spectrometer.
spec.ChIR_End	Ending point for the IR channel data in the spectrometer.
spec.IntLoc	Index of the maximum of the integrated intensity for the spectrometer.
spec.IntMax	Maximum of the integrated intensity in the spectrometer.
spec.intStart	Calculated start point that occurs above the threshold used to estimate the lag in the spectrometer data.
spec.intThreshold	Threshold for the spectrometer before ignition, measured by user input.
spec.int_time	Integration time of the spectrometer in $\mu$ s.
spec.molemission	User-entered channels to be selectively removed from consideration for temperature measurements with the spectrometer data.
spec.par	Parameters of the MC-PMT at the time of recording, imported using <code>importpar.m</code> .
spec.processedchdata	Channel data processed with the channel-specific MC-PMT gains applied.
spec.red_channeldata	Only channel data from the reduced spectrometer data.
spec.red_data	Reduced data matrix for the spectrometer with adjustments for pre-trigger samples.
spec.red_time	Time stamps for reduced channel data.
spec.sample_rate	Sample rate of the spectrometer at the time of recording in Hz.

<code>spec.time</code>	Timestamps of the spectrometer data to be used for data processing.
<code>spec.updprocessedchdata</code>	Processed channel data that has been filtered to have an acceptable number of points necessary for temperature fitting.
<code>spec.volt</code>	Voltage applied to the MC-PMT at the time of recording, used to determine gains to apply to the data.
<code>SpecFile</code>	Spectrometer file handle to be used for data importing.
<code>t_diff</code>	Difference in timestamps between the peak light in the photodiode and peak light in the channel of the MC-PMT with the peak wavelength sensitivity at the same point as the photodiode.
<code>t_diodemax</code>	Timestamp of the peak signal in the photodiode data.
<code>t_specmax</code>	Timestamp of the peak signal in the spectrometer channel with sensitivity at the same wavelength as the peak sensitivity of the photodiode.
<code>T.grey</code>	Estimated graybody temperature (in K) as a function of time.
<code>T.grey_err</code>	Error in the graybody estimated temperatures. Format is [Lowest temperature, Highest Temperature] with temperature being in K.
<code>T.grey_smooth</code>	Smoothed graybody temperatures (in K) as a function of time.
<code>T.residual</code>	Residual error in temperature measurement returned from <code>TempFit.m</code> .
<code>T.time</code>	Timestamps of the temperature points within the struct.
<code>T.transpose</code>	Estimated temperature (in K) using a transpose fit as a function of time. This data is most commonly used for publishing.
<code>T.transpose_err</code>	Error in the transpose-fit calculated temperatures. Format is [Lowest temperature, Highest Temperature] with temperature being in K.
<code>T.transpose_select</code>	Filtered <code>T.transpose</code> data where the calculated error is <350 K.
<code>T1</code>	Temporary variable used to improve visuals in checking for broken/sharp peaks.

T2	Temporary variable used to improve visuals in checking for broken/sharp peaks.
T3	Temporary variable used to improve visuals in checking for broken/sharp peaks.
temp	Temporary variable.
testCh	Test channels entered by user to check for the smoothing parameter in the spectrometer data.
XLocs	User input information recovered from graphical user interface inputs.

#### ***N.4. Code structure***

The pressure cell temperature measurement code is broken up into discrete sections as outlined in the code itself and separated by `%%`. A description of each of these sections and their main functions is below, but the program has been heavily commented for further reference.

**Table N-5:** Code structure for pressure cell temperature measurement code.

<b>Section</b>	<b>Description</b>
File Folder Information Import	Identifies folder with files enclosed to be used for data processing.
Pressure Data Import	Import of pressure data exported by the oscilloscope.
Diode Data Import	Import of the photodiode data exported by the oscilloscope.
Spectrometer – Data Import	Import of spectrometer data from the .txt file generated by the Log File Converter in the Vertilon software. Import routines call on external functions. Setup of the <code>spec</code> struct.
Spectrometer – Molecular Emission Check	Routine where users identify channels in the spectrometer data that are prone to molecular emission or inadvertent absorption. Channels with obstructions are recorded and excluded from temperature measurement.
Spectrometer – Smoothing Data	Routine where user identifies channels to check in the spectrometer data and then determines the smoothing parameter to apply to the channel data.



Spectrometer – Filtering Data	Data filtering routine where data above the noise level and below the linearity limit of the MC-PMT is chosen and the specific channel gains from the latest calibration are applied.
Spectrometer – Temperature Measurement	Temperature calculations are performed using the TempFit.m routine and stored in the T struct.
Spectrometer and Diode – Time Shifting	Routine that identifies the time shift between data collected in the photodiode and the data recorded by the spectrometer data acquisition unit.
Data Show – Pressure/Diode	Figure creation showing the pressure and diode traces.
Data Show – Pressure/Temperature	Figure creation showing the pressure and temperature traces.
Save Data	Creation of a new .mat data file in the working folder with the name of the folder containing all of the relevant data.

### ***N.5. Summary***

This user manual and code description includes the input, output, and structure of the program used to measure temperature in the constant volume combustion cell in the Zachariah Group. As previously mentioned, this code was only rewritten for user simplicity, but all of the routines and methods developed herein are part of work performed by and/or with Dr. Rohit Jacob. If sections of this code are used in other programs or should be cited as part of future research, please reference R.J. Jacob, D.J. Kline, M.R. Zachariah, High speed 2-dimensional temperature measurements of nanothermite composites, J. Appl. Phys. 123 (2018) 115902. doi:10.1063/1.5021890.

### ***N.6. MATLAB code***

```
%% Temperature Measurement based on Emission in a Constant Volume
Combustion Cell
% Dylan J. Kline
% Chemical Engineering – Zachariah Group
% University of Maryland College Park

% This code has been adapted from older versions of similar
applications
```

```

% by Dr. Rohit Jacob during his time in the Zachariah Group - his
efforts
% are greatly appreciated and this code would not exist without his
work.
% It has simply been rewritten here to make life a little easier.
The code
% is broken up into multiple sections as described below:
%

% Notes from Rohit.
%
% change molec emission channel: [1 5 18 22 23] for physically mixed
samples and WO3 and Fe2O3 meso
% CuO meso use [1 5 9 18 22 23]
% change ChIR selection criteria if peak pressure points are being
missed out for low performing systems
% change pressure prominence to 2 psi for low performance systems
else 5 psi (CuO meso)
%

clc
clear
close all
tic

% Adds all external functions to working path.
addpath(genpath(fileparts(which(mfilename))))

%% File Folder Information Import

FolderName = uigetdir; % folder containing the spectra
file converted to txt and all channel data
cd(FolderName) % go to folder in finder

OscFile = dir('C*.dat'); % gets information for all of the
channels (1,2,4 or pressure, diode, TTL);

if isempty(dir('*.mat'))==1 % Checks to see if data was
converted using log file converter or matvertilon
    SpecFile = dir('*.txt'); % gets information for
spectrometer file (log file converter)
    SpecFile = fullfile(pwd,SpecFile.name);
elseif isempty(dir('*.mat'))==0
    SpecFile = dir('*.mat');
    for k=1:size(SpecFile,2) % Needed to remove problem
with temporary files
        if SpecFile(k).name(1) == '.'
            SpecFile(k)=[];
        end
    end
    SpecFile = fullfile(pwd,SpecFile.name);
else
    disp('No spectrometer file');
    return;
end

%% Pressure Data Import
% [time, voltage, P(psi), P(kPa)]

disp('Importing pressure data...');
P.data = load(OscFile(1).name); % import pressure
data (channel 1)

```

```

P.data(P.data(:,1)<0,:)=[];
%P(:,2) = smooth(P(:,2),5,'moving'); % smooth pressure
data
P.data(:,2) = medfilt1(P.data(:,2),150); % trying to
smooth out the pressure data
P.data(:,2) = smooth(P.data(:,2),100,'moving'); % trying to
smooth out the pressure data
P.data(:,3) = P.data(:,2) * 1000/1.01 * 6.89476; % convert mV to
kPa

%% Diode Data Import
% [time, voltage, mA]

disp('Importing diode data...');
D.data = load(OscFile(2).name); % import diode
channel (channel 2)
D.data(D.data(:,1)<0,:)=[];
D.data(:,2) = -1*smooth(D.data(:,2),100,'moving'); % smooth diode
channel
D.data(:,3) = D.data(:,2) * 1000/50; % 50 ohm
termination convert to mA

%% Spectrometer - Data Import

disp('Importing spectrometer data...');
[chWave,chWid] = channelWavelengthGen(663,150); %
spectrometer CW and grating (1/mm)

if isempty(dir('*.mat'))==1
    spec.par = importpar(SpecFile); %
parameters for spectrometer
    spec.int_time = spec.par(1,1); % in
us
    spec.sample_rate = spec.par(3,2); %
sample rate (Hz)
    spec.volt = importvolt(SpecFile); %
voltage on spectrometer
    spec.alldata = importfile(SpecFile); %
import spectrometer data

    spec.channeldata = spec.alldata(:,6:37); %
import spectrometer data for channels only
    spec.time = spec.alldata(:,38) - spec.alldata(1,38); %
fixing some timestamps (i think)
else
    load(SpecFile,'config_user'); %
Imports user configuration header
    spec.sample_rate = config_user.TrigPeriod(1); %
Imports the trigger period
    spec.int_time = config_user.IntegPeriod(1)*1e6; %
Imports integration period time, converts to us
    spec.volt = config_user.HVSetpoint0; %
Imports voltage on spectrometer
    load(SpecFile,'data'); %
Loads data file
    spec.alldata = data; %
Loads data file
    spec.channeldata = spec.alldata(:,7:38); %
Imports channel data
    spec.time = spec.alldata(:,41); %
Unshifted time
    clearvars data %
clears data variable

```

```

end

%% Spectrometer - IR Channel Starting and Filtering
TimeStamp = spec.time; % Temporary timestamp
spec.ChIR = spec.channeldata(:,2); % Using channel 2 as an
IR channel
spec.ChIR = medfilt1(spec.ChIR); % Applying filter to
channel data
spec.ChIR = smooth(spec.ChIR,5,'moving'); % Smoothing IR data
DaqTimeStart = spec.time(1); % Not sure what this does
TimeStamp = TimeStamp(1)-DaqTimeStart;

[~,spec.ChIR_NoiseIndex] = max(spec.ChIR);
% Gets noise from data
spec.ChIR_Threshold =
round(1.5*max(spec.ChIR(1:round(0.85*spec.ChIR_NoiseIndex))),2);
% 50% higher than background noise
spec.ChIR_Start = find(spec.ChIR > spec.ChIR_Threshold,1);
% Point of starting for IR data.
spec.ChIR_End = (spec.ChIR_Start + 1000) +
find(spec.ChIR((spec.ChIR_Start + 1000):end) < spec.ChIR_Threshold,1)
- 1; % Ending point for IR window

preTriggerSamples = round(5e-3*spec.sample_rate); % amount of time
for pretrigger
if (spec.ChIR_Start-preTriggerSamples < 0) % If there IR
starting point is less than the number of pre trigger samples (rare)
linestart = 1; % read from
first data point.
else
linestart = spec.ChIR_Start-preTriggerSamples; % Otherwise,
read from the first point beyond the pre-trigger samples
end

% VARIABLES WITH RED_ BEFORE INDICATE REDUCED DATA FOR PROCESSING
spec.red_data =
spec.alldata(linestart:(spec.ChIR_End+preTriggerSamples),:); %
Reduced data, Unused data being removed
spec.red_channeldata = spec.red_data(:,7:38);
% Only channel data
spec.red_time = spec.red_data(:,41)-DaqTimeStart;
% Shifted time so first sample is 0

%% Spectrometer - Molecular Emission Check
% Creates a graph of the averaged emission recorded with the
spectrometer
% to determine whether or not there is molecular emission disturbing
the
% temperature measurement.

%%%%% CREATING A FIGURE %%%%%
[~,temp] = max(spec.red_channeldata(11:end,1));
f1 = figure('units','normalized','outerposition',[0 0 0.5 0.5]);
plot(1:32,mean(spec.red_channeldata(temp-
10:temp+10,1:32)), 'LineWidth',3); % average of 21 spectrum so as
to check the presence of any molecular emission
grid on;
ax1=gca;
ax1.Position = [0.1300 0.1100 0.7750 0.7150];
ax1.XTick = 1:32;
ax1.XLim = [1,32];
ax1.FontSize = 10;
ax1.XLabel.String = 'Channel';

```

```

ax1.XLabel.FontWeight = 'bold';

T1 = (num2str((chWave(1,:) '*1e9')));
T2 = (num2str((chWave(2,:) '*1e9')));
T3 = [T1, repmat(' - ', 32, 1), T2];
C = fliplr(cellstr(T3)');
clear T1 T2 T3;

pos = [0.1300    0.1100    0.7750    0.750];
ax2 =
axes('Position',pos,'XAxisLocation','top','YAxisLocation','right','Color','none','XLim',[chWave(3,end) chWave(3,1)]*1e9);
ax2.YTick = [];
ax2.XTick = fliplr(chWave(3,:)*1e9);
ax2.XDir = 'reverse';
ax2.XTickLabel = C;
ax2.XTickLabelRotation = 45;
ax2.FontSize = 10;
ax2.XLabel.String = 'Wavelength (nm)';
ax2.XLabel.FontWeight = 'bold';
%%%%% END FIGURE CREATION %%%%%

spec.molemission = input(sprintf('Are there any sharp peaks/ broken
points that are prominent in the spectra?\nEnter the X-axis positions
as an array: '));
close(f1);

%% Spectrometer - Smoothing Data
% This will check whether or not the spectra are properly smooth. A
typical parameter is ~5.
% Channels to be checked for smoothing should be ones that were
% eliminated.

r = size(spec.red_channeldata,1);
NumChan = size(spec.red_channeldata,2);

%%%%% CREATING A FIGURE, SMOOTHING %%%%%
CurrentLimit = 25;
% in uA
%CurrentLimit = 30;
ChargeLimit = CurrentLimit*spec.int_time;
% in pC
NaNCounter = zeros(r,1);

CD = spec.red_channeldata;
testCh = input('Enter 3 channels as an array to test smoothing: ');
%[5,15,22] for Bi, [5,22,30] for Cu,W and Fe

f1 = figure('units','normalized','outerposition',[0 0 0.5 0.5]);
h1 =
plot(1:r,spec.red_channeldata(:,testCh(1)),1:r,spec.red_channeldata(:,testCh(2)),1:r,spec.red_channeldata(:,testCh(3)));
ylim([0 2*max(spec.red_channeldata(:,1))]);
hold on;
title('Improve Smoothing?');
check = input('Improve Smoothing? (0/1): ');
while check
    Smoother = input('Enter smoothing parameter (1-50): ');
    for i = 1:NumChan
        CD(:,i) =
smooth(spec.red_channeldata(:,i),Smoother,'moving');
    end
end

```

```

        h2 =
plot(1:r,CD(:,testCh(1)),1:r,CD(:,testCh(2)),1:r,CD(:,testCh(3)),'Lin
eWidth',2);

        check = input('Improve Smoothing? (0/1): ');
        if check
            delete(h2);
        end
    end
close(f1);
%%%%% END CREATING A FIGURE, SMOOTHING %%%%%
spec.red_channeldata = CD;
clearvars CD;

% Baseline Selection
% Ideally, the line should be around where the noise lies

%%%%% CREATING A FIGURE, CHECKING %%%%%
% start the baseline edits
Baseline = 0.1;

f1 = figure('units','normalized','outerposition',[0 0 0.5 0.5]);
plot(spec.red_channeldata);
hold on;
plot([1,size(spec.red_channeldata,1)],repmat(Baseline,1,2),[1,size(sp
ec.red_channeldata,1)],repmat(ChargeLimit,1,2),'LineWidth',4);
set(gca,'FontSize',20);
%yyaxis right;
%plot(diff(spec.time),'.r');
title('Is Baseline OK');
ylim([0 inf])
check = input('Is baseline ok? (0/1): ');
while ~check
    Baseline = input('New baseline: ');
    %yyaxis left;

plot([1,size(spec.red_channeldata,1)],repmat(Baseline,1,2),'LineWidth
',4);
    check = input('Is baseline ok? (0/1): ');
end
close(f1);
%%%%% END CREATING A FIGURE, CHECKING %%%%%

% Spectrometer - Filtering Data
% Filtering data above the current limit and below the baseline
[LimRow,LimCol] = find(spec.red_channeldata <= Baseline |
spec.red_channeldata > ChargeLimit);

% Loading in the correction factor gains for the spectrometer based
on the
% last calibration
load('PcellCorrFactorND2_20200707_DJK.mat')

Gains = PCellCorrFactorND2(PCellCorrFactorND2(:,1)==spec.volt,3:34);
% extracting the channel gains based on the voltage used
CalibIntTime =
PCellCorrFactorND2(PCellCorrFactorND2(:,1)==spec.volt,2); %
CalibIntTime in us

spec.processedchdata =
bsxfun(@times,spec.red_channeldata(:,1:size(Gains,2)),Gains); %
correcting the channels with the gains

```

```

Radiance = spec.processedchdata;
% for plotting
Radiance(Radiance<0) = 0;
% fixing the possibility that the radiance could be negative
IntegratedIntensity = sum(Radiance(:,1:32),2);
% sum(Radiance(Radiance<0)=0) also gives same results

% Removing data for the current limit
for i = 1:size(LimRow,1)
    spec.processedchdata(LimRow(i),LimCol(i)) = NaN; % get rid of
all OR (Out of Range) data points
end
% Removing data for molecular emission
spec.processedchdata(:,spec.molemission)= NaN; %
for molecular emission

% Removing points with too many NaN's
if isempty(dir('*.mat'))==1
    err_oor = 3;
    err_inp = 4;
else
    err_oor = 4;
    err_inp = 5;
end
for i = 1:r %
Count the number of NaN in each row so as to know which were removed
    if spec.alldata(r,err_oor) == 1 || spec.alldata(r,err_inp) == 1 %
        spec.processedchdata(r,:) = NaN;
    end
    If file shows Input Error or Out of Range
    end
    NaNcounter(i) = sum(isnan(spec.processedchdata(i,:)));
end

spec.processedchdata = [spec.processedchdata spec.red_time];
% need time stamp when getting rid of rows with NaN > threshold

j = 1;
for i = 1:r %
    if NaNcounter(i)<21 %
21 is the threshold of NaN's in array
        spec.updprocessedchdata(j,:) = spec.processedchdata(i,:);
        j = j+1;
    end
end

ProcRow = size(spec.updprocessedchdata,1); %
updated size of ProcessedChData

%% Spectrometer - Temperature Measurement
% Uses an external function to estimate the temperature using the
graybody
% method and transposed temperature method. Typcially, the
transposed
% temperature would be okay, but it is prone to having high errors
and so
% the graybody is provided for thoroughness.

T.transpose = zeros(ProcRow,1);
T.transpose_err = zeros(ProcRow,2);
T.grey = zeros(ProcRow,1);
T.grey_err = zeros(ProcRow,2);
T.residual = zeros(ProcRow,1);

```

```

AbsIntensityData = zeros(ProcRow,NumChan);

%%% The code has the tendency to generate a lot of errors %%%
warning('off','all');
wait = waitbar(0,'Estimating temperature');

for i = 1:ProcRow

    AbsIntensityData(i,:) =
spec.updprocessedchdata(i,1:NumChan)*CalibIntTime/(spec.int_time*chWi
d);

    normCh = find(~isnan(spec.updprocessedchdata(i,1:32))==1,1);

    Output =
TempFit(AbsIntensityData(i,:),chWave(3,1:NumChan),normCh,2); % call
to external function returns a variable output

    T.grey(i) = Output.Greybody_temp;
% storing the greybody temperature
    T.transpose(i) = Output.Transpose_temp;
% storing the transposed calculated temperature

    T.residual(i) = Output.ResNorm;
% storing the residual
    T.grey_err(i,1) = T.grey(i)-min(Output.Greybody_Error);
% storing the greybody negative error
    T.grey_err(i,2) = max(Output.Greybody_Error)-T.grey(i);
% storing the greybody positive error
    T.transpose_err(i,1) = T.transpose(i)-min(Output.Error);
% storing the transpose negative error
    T.transpose_err(i,2) = max(Output.Error)-T.transpose(i);
% storing the transpose positive error

% Output.Error is already formatted as [max,min]
    %disp(i);
    waitbar(i/ProcRow,wait,'Estimating temperature');
end
close(wait)

%%% Turning back on the error log %%%
warning('on','all');

T.grey(T.residual > 0.5) = NaN; % use only data if
residual less than 0.5 (random)
T.grey_err(T.residual > 0.5,:) = NaN; % cropping error data
down appropriately
T.grey_smooth = smooth(T.grey,3,'moving'); % smoothing the
greybody temperature

T.transpose_select = T.transpose; % Change smoothing
parameter for temperature here
T.transpose_select(T.transpose_err(:,1)>350 |
T.transpose_err(:,2)>350) = NaN; % removing transposed temperature
data with high error

T.time = spec.updprocessedchdata(:,end);

%% Spectrometer and Diode - Time Shifting (NO LONGER USED)

```



```

% It has become quite apparent that the time signatures of the diode
and
% spectrometer are off slightly because of the effective length of
the
% light path. As such, the peak light intensity will be used to
correct
% the spectrometer light signal up to the time of the diode light
signal.
% The photodiode is most sensitive to light ~495 nm, so it would be
best
% to time shift in accordance with that channel on the spectrometer.

% [~,idx_diodemax] = nanmax(D(:,3));
% t_diodemax = D(idx_diodemax);
%
% [~,idx_specmax] = nanmax(IntegratedIntensity);
% [~,idx_specmax] = nanmax(Radiance(:,30));
% t_specmax = spec.time(idx_specmax);
%
% t_diff = t_diodemax - t_specmax;
%
% T.time = T.time + t_diff;

%% Integrated Intensity Data Window (Spectrometer)
[spec.IntMax,spec.IntLoc] = max(IntegratedIntensity); %
Retrieve maximum of integrated intensity

% Routine to have user select the end of the noise window in the 32
% channel integrated spectrometer
f1 = figure('units','normalized','outerposition',[0 0 0.5 0.5]);
plot(IntegratedIntensity);
title('Please select the point to end the noise window for 32Ch
Integrated','FontSize',20);
[XLocs,~] = ginput(1);
close (f1);

XLocs = round(XLocs); %
Round point for end of noise window
spec.intThreshold = round(1.1*max(IntegratedIntensity(1:XLocs)),3); %
integrated lag threshold for the spectrometer
spec.intStart = find(IntegratedIntensity>spec.intThreshold,1); %
start point for the spectrometer to adjust lag

%% Pressure Data Window
Buffer = -0.5e-2;
% To ensure all pretrigger data from Diode signal is imported
 [~,P.ImportStart] = min(abs(P.data(:,1)-spec.red_time(1)+Buffer));
% Starting point for reduced pressure signal data considering the lag
time
% [~,P.ImportEnd] = min(abs(P.data(:,1)-spec.red_time(end)-Buffer));
% Ending point for reduced pressure signal data considering the lag
time
P.Import =
[P.data(P.ImportStart:P.ImportEnd,1),P.data(P.ImportStart:P.ImportEnd
,3)];% Concatenating data to a new variable for pressure data
importing
P.Import = [P.data(P.ImportStart:end,1),P.data(P.ImportStart:end,3)];
% Concatenating data to a new variable for pressure data importing

% Routine to have user find end of noise range for pressure window.
f1 = figure('units','normalized','outerposition',[0 0 0.5 0.5]);
plot(P.Import(:,2));

```

```

title('Please select the point to end the noise window for
Pressure','FontSize',20);
[XLocs,~] = ginput(1);
close (f1);

XLocs = round(XLocs);
if sum(P.Import(1:XLocs,2)>0)
    P.Threshold = round(1.1*max(P.Import(1:XLocs,2)),4);
    P.StartLoc = find(P.Import(:,2)>P.Threshold,1);
else
    P.Threshold = round(0.9*max(P.Import(1:XLocs,2)),4);
    P.StartLoc = find(P.Import(:,2)>P.Threshold,1);
end

%% Diode Data Window

[~,D.ImportStart] = min(abs(D.data(:,1)-spec.red_time(1)+Buffer));
% Starting point for reduced diode signal data considering the lag
time
[~,D.ImportEnd] = min(abs(D.data(:,1)-spec.red_time(end)-Buffer));
% Ending point for reduced pressure signal data considering the lag
time
%D.Import = [D.data(D.ImportStart:D.ImportEnd,1),
D.data(D.ImportStart:D.ImportEnd,3)]; % Concatenating data to a new
variable for diode data importing
D.Import = [D.data(D.ImportStart:end,1),
D.data(D.ImportStart:end,3)]; % Concatenating data to a new variable
for diode data importing

% Routine to have user find end of noise range for pressure window.
f1 = figure('units','normalized','outerposition',[0 0 0.5 0.5]);
plot(D.Import(:,2));
title('Please select the point to end the noise window for
Diode','FontSize',20);
[XLocs,~] = ginput(1);
close (f1);

XLocs = round(XLocs);
D.Threshold = round(1.1*max(D.Import(1:XLocs,1)),5);
D.StartLoc = find(D.Import(:,2)>D.Threshold,1);

%% Pressure and Diode Lag Calculations
T.time = spec.updprocessedchdata(:,NumChan+1); %
Correcting time for temperature
P.intLag = P.Import(P.StartLoc,1)-spec.red_time(spec.intStart); %
Integrated lag time for pressure in data showing
%D.intLag = D.Import(D.StartLoc,1)-spec.red_time(spec.intStart); %
D.intLag = P.intLag; %
Making pressure and diode lag time the same since they should
realistically be
T.time = (T.time-spec.red_time(spec.intStart))*1000; %
make time zero to be the int start in ms
% RadianceTime = (spec.red_time-spec.red_time(spec.intStart))*1000;
% Time for intensity data shifted to intstart, not used
% P.Lag = min(85e-6,P.intLag);
% disp(['Effective Pressure Lag used: ',num2str(P.Lag),' s']);

%% Final Data to Show
DataFinal.P = [(P.Import(:,1)-P.intLag-
spec.red_time(spec.intStart))*1000, (P.Import(:,2)-
mean(P.Import(1:P.StartLoc,2)))]];

```

```

DataFinal.D = [(D.Import(:,1)-D.intLag-
spec.red_time(spec.intStart))*1000, D.Import(:,2)];
DataFinal.T = [T.time,
T.transpose,T.transpose_select,T.transpose_err,T.grey_smooth,T.grey_e
rr];

%% Data Show - Pressure/Diode

%%%%%% CREATING A FIGURE, CHECKING %%%%%%
fig=figure('Color',[1 1 1]);
set(fig,'defaultAxesColorOrder',[0 0 0]; [0 0 0]);
hold on
box on

yyaxis left
ax1 = plot(DataFinal.P(:,1),DataFinal.P(:,2),'r','LineWidth',1.25); %
converting to kPa
ylabel('Pressure (kPa)');

yyaxis right
ax2 = plot(DataFinal.D(:,1),DataFinal.D(:,2),'b','LineWidth',1.25);

legend('Pressure','Diode')

ylabel('Diode (arb.)')
xlabel('time (ms)')

%xlim([0, inf]);
ax=gca;
ax.FontSize=14;
set(fig,'defaultAxesColorOrder',[0 0 0]; [0 0 0]);
hold off
%%%%%% END CREATING A FIGURE %%%%%%

%% Data Show - Pressure/Temperature

%%%%%% CREATING A FIGURE, CHECKING %%%%%%
fig2=figure('Color',[1 1 1]);
set(fig2,'defaultAxesColorOrder',[0 0 0]; [0 0 0]);
hold on
box on

yyaxis left
ax3 = plot(DataFinal.P(:,1),DataFinal.P(:,2),'r','LineWidth',1.25); %
converting to kPa
ylabel('Pressure (kPa)');

yyaxis right
ax4 =
errorbar(DataFinal.T(:,1),DataFinal.T(:,3),DataFinal.T(:,4),DataFinal
.T(:,5));
%ax4 =
errorbar(T.time*1000,T.transpose_select,T.transpose_err(:,1),T.transp
ose_err(:,2));
set(ax4,'Marker','o','MarkerSize',8,'MarkerEdgeColor','b','MarkerFace
Color','w','CapSize',0,'LineWidth',0.2)
%ax4 = plot(T.time*1000,T.grey_smooth,'b','LineWidth',1.25);

ylabel('Temperature (K)')
xlabel('time (ms)')

```

```

legend('Pressure','Temperature (Transposed)')

%xlim([0, inf]);
ax=gca;
ax.FontSize=14;
set(fig2,'defaultAxesColorOrder',[0 0 0]; [0 0 0]);
hold off
%%%%%%%% CREATING A FIGURE, CHECKING %%%%%%%%%

%% Save Data
[~,name,~] = fileparts(FolderName);
save(sprintf('%s.mat',strrep(name,'
','_')),'T','D','P','spec','DataFinal');

```

## Appendix O: MatVertilon – a multiplatform MATLAB program to extract data from Vertilon “.log” data files

### ***O.1. Introduction***

A functionalized MATLAB program has been developed for the import of the “.log” video format that the Vertilon IQSP family uses for saving their data. Prior to writing this software, data conversion would take multiple hours per file that was analyzed and would only work on a Windows machine when plugged into the actual device, thus rendering the computer that was used for data acquisition unusable for extended periods of time. Similar to the MatCine program, this was a solution to this problem and has also reduced the time to ~10 minutes per file (a substantial improvement over the many-hour alternative). Detailed .log file format developer documents can be provided by request to the original equipment manufacturer (Vertilon).

### ***O.2. Running MatVertilon***

The program required input is a .log data file. The program then executes and converts the data to a MATLAB format that can be imported into other processing routines.

The last edit of this software was on October 5, 2020. If sections of this code are used in other programs or should be cited as part of future research, please reference this dissertation. An online version has also been stored with version control and is available by reasonable request.

### 0.3. MATLAB code

```
function [id, config_header, config_user, config_factory,
data]=MatVertilon(filename)

%% Vertilon Log File Importer

% Dylan J. Kline
% University of Maryland College Park
% dkline@umd.edu
%
% This program was written during the great coronavirus shutdown
after
% many many years of struggling with the LogFileConverter program
that
% Vertilon made (it isn't multiplatform compatible and isn't
standalone).
% The log files generated by are so packed with different types of
data
% that it was difficult to really justify going through each
variable and
% making sure that it was decoded properly. Surely, if it is
necessary to
% have this extraneous data in the right format, then it would be
just a
% matter of making sure I decoded everything in the right format.
%
% Data should be a .log file exported by the Vertilon Data
Acquisition
% Software during spectrometry measurements using a IQSP DAQ. In the
% Zachariah Group (UCR), we have a IQSP580. Data formats and the
program
% structuring were determined using the user manual for the device
as of
% 06/03/2020.

%clear
tic
% Open log file
% logfile = fopen('20190306_500V_5ms.log');
logfile_info = dir(filename);
% logfile = fopen('nSi CuO run1.log');
% logfile_info = dir('nSi CuO run1.log');
logfile_info = dir(filename);
logfile = fopen(filename);

word_len = 2; % 2 bytes, 16 bits
id_offset = 0; % Beginning of file
config_offset = 32; % Offset to beginning of config table
section (words)
config_user_offset = 33; % Offset to beginning of user config
table section (words)
config_factory_offset = 1283; % Offset to beginning of factory
config table section (words)
data_offset = 2033; % Offset to beginning of data section
(words)

%% ID Text Header
clearvars id
fseek(logfile,0,'bof'); % go to beginning of
file
loc = ftell(logfile);
```

```

id.ProductID      = fscanf(logfile,'%c',17);      % Product ID, 17
bytes, "Vertilon xxxxxx[CR][LF]"
id.DateTimeString = fscanf(logfile,'%c',19);      % Date/time string,
19 bytes, "MM/DD/YY HH:MM xx[CR][LF]"
id.UIVersion      = fscanf(logfile,'%c',28);      % Software UI
version, 28 bytes, "LabView UI Version xxxxxxxx[CR][LF]"

loc = ftell(logfile);

%% Config Header
clearvars config_header
fseek(logfile,config_offset*word_len,'bof');      % go to beginning
of config table section
loc = ftell(logfile);                             % location check

% Standard config table
config_header.major_rev = fread(logfile,1,'uint8','b'); % Config
table revision, 1st 8 bits = Major Rev
config_header.minor_rev = fread(logfile,1,'uint8','b'); % Config
table revision, 2nd 8 bits = Minor Rev

% location storer
a = 1;
loc(a) = ftell(logfile);
a=a+1;

%% User Configuration Table
clearvars config_user
fseek(logfile,config_user_offset*word_len,'bof'); % go to beginning
of config table section
loc = ftell(logfile);

config_user.SystemMode = fread(logfile,1,'int16','b'); % 0:
0=standby mode, 1=acquire mode

config_user.HVLimit0 = fread(logfile,1,'int16','b')/10; % 1:
Maximum voltage on HV0
config_user.HVLimit1 = fread(logfile,1,'int16','b')/10; % 2:
Maximum voltage on HV1

config_user.NumChannelsB0 = fread(logfile,1,'int16','b'); % 3:
Number of channels in bank 0
config_user.NumChannelsB1 = fread(logfile,1,'int16','b'); % 4:
Number of channels in bank 1
config_user.NumChannelsB2 = fread(logfile,1,'int16','b'); % 5:
Number of channels in bank 2
config_user.NumChannelsB3 = fread(logfile,1,'int16','b'); % 6:
Number of channels in bank 3

% I really hate vertilon for this. They have stored the information
as bits
% that need to be imported in binary format and read individually.
temp = dec2bin(fread(logfile,1,'int16','b'));
config_user.HV0Enabled = str2num(temp(1)); % 7:
0=disable
if str2num(temp)>10
    config_user.HV1Enabled = str2num(temp(2)); %
else
    config_user.HV1Enabled = 0; %
end

```

```

config_user.HVSetpoint0 = fread(logfile,1,'int16','b')/10;      % 8:
Voltage setpoint on HV0
config_user.HVSetpoint1 = fread(logfile,1,'int16','b')/10;      % 9:
Voltage setpoint on HV1

config_user.UserConfigID = fread(logfile,1,'int16','b');        % 10:
Unused
config_user.DCRD_AOut_0 = fread(logfile,1,'int16','b');        % 11:
0-4095, 3.0V full scale

% 12: Matrix filtering band enables, 0-255 (1 bit per band)
config_user.BandEnables = fread(logfile,1,'int16','b');

% 13-28: Band "J-1" Index, [Start End], words
for j=1:8
    for k=1:2
        config_user.BandStartEndIndex(j,k) =
fread(logfile,1,'int16','b');
    end
end

% 29: 0-255, each bit position corresponds to a flag enable
config_user.FlagEnables = fread(logfile,1,'int16','b');

% 30-61: Flag "J-1" Operand "K", [Operand0 Operand1 Operand2
Operand3]
for j=1:8
    for k=1:4
        config_user.FlagOperand(j,k) = fread(logfile,1,'int16','b');
    end
end

% 62-69: MPterm "J-1"
for j=1:8
    config_user.PTerm(j) = fread(logfile,1,'int16','b');
end

config_user.DataFilterEnable = fread(logfile,1,'int16','b');    % 70:
Spectral filtering data filter blocks data output if there is no
spectral filter match, 0=disabled

temp = dec2bin(fread(logfile,1,'int16','b'));
config_user.ProcessingEnables_SpectralFiltering = str2num(temp(1));
% 71: 0=disable
config_user.ProcessingEnables_BackgroundSubtract = str2num(temp(2));
%
config_user.ProcessingEnables_BackgroundSubtract = str2num(temp(2));
%

config_user.TimestampEnable = fread(logfile,1,'int16','b');      % 72:
enables/Disables timestamp output, 0=disable
config_user.DAC_Spare = fread(logfile,1,'int16','b');            % 73:
SIB analog out control 0-4095 (3.0V full scale)
config_user.TimestampInterval = fread(logfile,1,'int','b');      % 74-
75: Timestamp interval configuration, RAnge 10-100000 (10ns per bit)
config_user.CustomWordsEnable = fread(logfile,1,'int16','b');    % 76:
enables/Disables custom word output, 0=disable
config_user.EventCustomCount = fread(logfile,1,'int16','b');     % 77:
Number of custom words, range 0-64
config_user.RESERVED = fread(logfile,1,'int16','b');             % 78:
Unused

```



```

config_user.ImageAcqMode = fread(logfile,1,'int16','b');           % 79:
0=particle, 1=image
config_user.InputTrigThresh = fread(logfile,1,'int16','b');       % 80:
Input trigger threshold, range 1-8191
config_user.InputTrigChannel = fread(logfile,1,'int16','b');      % 81:
Input trigger threshold, range 1-8191
config_user.RangeErrorEnable = fread(logfile,1,'int16','b');      % 82:
Enables/disables range and error output
config_user.CrossBankConfig = fread(logfile,1,'int16','b');       % 83:
current cross-bank configuration, bit0=cross bank enable, b1=bank1
main trigger, b2=bank2 main trigger...
config_user.ReportPackingMode = fread(logfile,1,'int16','b');     % 84:
0=real time acquisition (no packing), 1=high speed acquisition
config_user.GPOutputEnable = fread(logfile,1,'int16','b');        % 85:
general purpose output, 0=GP Out disabled, 1= Always on, 2=spectral
filter match only
config_user.GPOutputDelay = fread(logfile,1,'int','b');           % 86-
87: general purpose output delay, range 10-200000 (0.1-2000us)
config_user.GPOutputPeriod = fread(logfile,1,'int','b');          % 88-
89: general purpose output period, range 10-200000 (0.1-2000us)
config_user.IntBoxcarEnable = fread(logfile,1,'int16','b');       % 90:
Enables/disables boxcar mode, 0=disabled
config_user.BoxcarWidthEnable = fread(logfile,1,'int16','b');     % 91:
Enables/disables boxcar width output, 0=disabled

% 92-99: Unused
for j=0:3
    config_user.ResetDelay(j+1) = fread(logfile,1,'int','b');
end

% 100-103: 0=external, 1=internal, 2=level, 3=input, 4=dsp, 5=pre
for j=0:3
    config_user.TrigSource(j+1) = fread(logfile,1,'int16','b');
end

% 104-111: Range 500 - 10000000 (200kHz-10Hz)
% Check this out, you have to read this in as a binary value, flip
the
% first 16 and last 16 bytes, convert it back to a decimal number,
and then
% divide 1e8 by that number.
for j=0:3
    temp = dec2bin(fread(logfile,1,'uint','b'),32); % reading in as
binary value
    temp = horzcat(temp(17:end),temp(1:16)); % swapping bytes
    config_user.TrigPeriod(j+1) = 1e8/bin2dec(temp);
end

% 112-119: Range 5 - 10000000 (0.05-100000us)
% Yet another trip here... so you have to read in these values in a
weird
% way. You read in 16 bytes at a time and then need to flip the two
strings of 16,
% concatenate them, then read the value and do the conversion.
for j=0:3
    temp = dec2bin(fread(logfile,1,'uint','b'),32); % reading in
as binary value
    temp = horzcat(temp(17:end),temp(1:16)); % swapping
bytes
    config_user.IntegPeriod(j+1) = bin2dec(temp)*1e-8; % rereading
values, converting to s
end

```

```

% 120-127: Range 5 - 10000000 (0.05-1000000us)
for j=0:3
    temp = dec2bin(fread(logfile,1,'uint','b'),32);           % reading
    in as binary value
    if length(temp)>1
        temp = horzcat(temp(17:end),temp(1:16));           % swapping
    bytes
        config_user.IntegDelay(j+1) = bin2dec(temp)*1e-8; %
    rereading values, converting to s
    end
end

config_user.SibSel(1) = fread(logfile,1,'int16','b');         %
128: Hamamatsu R5900U-L16
config_user.SibSel(2) = fread(logfile,1,'int16','b');         %
129: Hamamatsu H8711
config_user.SibSel(3) = fread(logfile,1,'int16','b');         %
130: Pacific Silicon Sensor AD-LA-16-9-DIL18
config_user.SibSel(4) = fread(logfile,1,'int16','b');         %
131: Hamamatsu H7260
config_user.SibSel(5) = fread(logfile,1,'int16','b');         %
132: Undefined
config_user.SibSel(6) = fread(logfile,1,'int16','b');         %
133: Reserved
config_user.SibSel(7) = fread(logfile,1,'int16','b');         %
134: Reserved
config_user.SibSel(8) = fread(logfile,1,'int16','b');         %
135: Reserved

config_user.TriggerEndCount = fread(logfile,1,'int','b');     %
136-137: Number of triggers allowed in acquire mode
config_user.TrigStampSelect = fread(logfile,1,'int16','b');   %
138: Triggerstamp enable, 0=disabled

% 139-142: 0: 17bit sign magnitude, 1: 16bit 2's comp w/shift (FS),
% 2: 16bit 2's comp no shift (HS)
for j=0:3
    config_user.DataFormat(j+1) = fread(logfile,1,'int16','b');
end

fseek(logfile,7*2,0);

% 150-405: Gain compensation values for each channel, range 0-
% 0xFFFFFFFF
for j=150:405
    config_user.ChGainComp(j-150+1,1) = fread(logfile,1,'int16','b');
end

% 406-661: Input triggering threshold for each channel, range 0-
% 0xFFFFFFFF
for j=406:661
    config_user.ChTrigThresh(j-406+1,1) =
    fread(logfile,1,'int16','b');
end

% 662-677: Input triggering enables bit packed for each channel,
% 0=disabled, bit packed for each channel
for j=662:677
    config_user.ChTrigEnable(j-662+1,1) =
    fread(logfile,1,'int16','b');
end

```

```

% 678: Matrix filtering band enables, 0-255 (1 bit per band)
config_user.MBandEnables = fread(logfile,1,'int16','b');

% 679-694: M Band "J-1" Index, [Start End], words
for j=1:8
    for k=1:2
        config_user.MBandStartEndIndex(j,k) =
fread(logfile,1,'int16','b');
    end
end

% 695: 0-255, each bit position corresponds to a flag enable
config_user.MFlagEnables = fread(logfile,1,'int16','b');

% 696-727: M Flag "J-1" Operand "K", [Operand0 Operand1 Operand2
Operand3]
for j=1:8
    for k=1:4
        config_user.MFlagOperand(j,k) = fread(logfile,1,'int16','b');
    end
end

% 728-735: MPterm "J-1"
for j=1:8
    config_user.MPterm(j) = fread(logfile,1,'int16','b');
end

config_user.MDataFilterEnable = fread(logfile,1,'int16','b'); %
736: Spectral filtering data filter blocks data output if there is no
spectral filter match, 0=disabled
config_user.MDataFilterConfig = fread(logfile,1,'int16','b'); %
737: Matrix A/B combine parameters
config_user.MDataFilterAChannels = fread(logfile,1,'int16','b'); %
738: Matrix A channel span in GUI
config_user.MDataFilterBChannels = fread(logfile,1,'int16','b'); %
730: Matrix B channel span in GUI
config_user.MDataFilterA = fread(logfile,1,'int16','b'); %
740: Matrix A parameters in row/column format
config_user.MDataFilterB = fread(logfile,1,'int16','b'); %
741: Matrix B parameters in row/column format
config_user.DisplaySetting = fread(logfile,1,'int16','b'); %
742: Display mode for GUI graphs
config_user.Bar32Channels = fread(logfile,1,'int16','b'); %
743: Channels for bar 32 graph
config_user.Bar64Channels = fread(logfile,1,'int16','b'); %
744: Channels for bar 64 graph
config_user.Bar128Channels = fread(logfile,1,'int16','b'); %
745: Channels for bar 128 graph
config_user.Bar256Channels = fread(logfile,1,'int16','b'); %
746: Channels for bar 256 graph
config_user.S8x8Channels = fread(logfile,1,'int16','b'); %
747: Channels for single 8x8 graph
config_user.D4x4ChannelsA = fread(logfile,1,'int16','b'); %
748: Channels for dual 4x4 graph A
config_user.D4x4ChannelsB = fread(logfile,1,'int16','b'); %
749: Channels for dual 4x4 graph B
config_user.D8x8ChannelsA = fread(logfile,1,'int16','b'); %
750: Channels for dual 8x8 graph A
config_user.D8x8ChannelsB = fread(logfile,1,'int16','b'); %
751: Channels for dual 8x8 graph B
config_user.S16x16Channels = fread(logfile,1,'int16','b'); %
752: Channels for single 16x16 graph

```

```

config_user.Bar32Attributes = fread(logfile,1,'int16','b');      %
753: Attributes for bar 32 graph
config_user.Bar64Attributes = fread(logfile,1,'int16','b');      %
754: Attributes for bar 64 graph
config_user.Bar128Attributes = fread(logfile,1,'int16','b');     %
755: Attributes for bar 128 graph
config_user.Bar256Attributes = fread(logfile,1,'int16','b');     %
756: Attributes for bar 256 graph
config_user.S8x8Attributes = fread(logfile,1,'int16','b');       %
757: Attributes for single 8x8 graph
config_user.D4x4Attributes = fread(logfile,1,'int16','b');       %
758: Attributes for dual 4x4 graph
config_user.D8x8Attributes = fread(logfile,1,'int16','b');       %
759: Attributes for dual 8x8 graph
config_user.S16x16Attributes = fread(logfile,1,'int16','b');     %
760: Attributes for single 16x16 graph

fseek(logfile, (1250-761)*2,0);
loc(2) = ftell(logfile);

%% Factory Configuration Table
clearvars config_factory
fseek(logfile,config_factory_offset*word_len,'bof');             % go to
beginning of config table section
loc = ftell(logfile);

config_factory.DSPRevCode = fread(logfile,1,'int','b');           % 1250-
1251: DSP Rev Code
config_factory.FPGARevCode = fread(logfile,1,'int','b');         % 1252-
1253: DSP Rev Code

% 1254-1509: DSP calculated background for each channel, range 0-
0xFFFF
for j=1254:1509
    config_factory.ChBckgndOffset(j-1254+1,1) =
fread(logfile,1,'int16','b');
end

% 1510-1765: DSP calculated electrical offset for each channel, range
0-0xFFFF
for j=1254:1509
    config_factory.ChElecOffset(j-1254+1,1) =
fread(logfile,1,'int16','b');
end

config_factory.SiteSerNum = fread(logfile,1,'int','b');           % 1766-
1767: Unused
config_factory.BoardSerNum = fread(logfile,1,'int','b');         % 1768-
1769: Unused
config_factory.SIBSpareControl = fread(logfile,1,'int16','b');   % 1770:
Unused
config_factory.SpeedDyRange = fread(logfile,1,'int16','b');      % 1771:
Speed dynamic range for each bank, nibble based
config_factory.HVPopulated0 = fread(logfile,1,'int16','b');      % 1772:
High voltage supply populated, 0=unpopulated
config_factory.HVPopulated1 = fread(logfile,1,'int16','b');      % 1773:
High voltage supply populated, 0=unpopulated
config_factory.BiasVoltage = fread(logfile,1,'int16','b');       % 1774:
Bias voltage control 1 DAC 1, 0-4095 (3.0V full scale)
config_factory.DREVoltage0 = fread(logfile,1,'int16','b');       % 1775:
0-4095 (3.0V full scale)
config_factory.Reserved = fread(logfile,1,'int16','b');          % 1776:
0-4095 (3.0V full scale)

```

```

% 1777-1780: Reset low threshold for bank 1 to bank 4, range 0-0xFFFF
for j=1777:1780
    config_factory.ResetLowThresh(j-1777+1,1) =
fread(logfile,1,'int16','b');
end

% 1781-1784: Reset high threshold for bank 1 to bank 4, range 0-
0xFFFF
for j=1781:1784
    config_factory.ResetHighThresh(j-1781+1,1) =
fread(logfile,1,'int16','b');
end

% 1785-1788: Out of range low threshold for bank 1 to bank 4, range
0-0xFFFF
for j=1785:1788
    config_factory.OORLowThresh(j-1785+1,1) =
fread(logfile,1,'int16','b');
end

% 1789-1792: Out of range high threshold for bank 1 to bank 4, range
0-0xFFFF
for j=1789:1792
    config_factory.OORHighThresh(j-1789+1,1) =
fread(logfile,1,'int16','b');
end

% 1793-1794: Test voltages for DAC 2 and DAC 3, 0-4095 (3.0V full
scale)
for j=1793:1794
    config_factory.VBTest(j-1793+1,1) = fread(logfile,1,'int16','b');
end

% 1795-1798: Channel processing enables for bank 1 to bank 4
% [Deserializer, Reset Threshold, Buffer, Difference Raw/Subtracted,
Offset, Gain, Range Adjust, Data Trigger]
for j=1795:1798
    temp = dec2bin(fread(logfile,1,'int16','b'));
    for k=1:8
        config_factory.ChProcessingEnables(j-1795+1,k) =
str2num(temp(k));
    end
end

% 1799-1802: Number of channels populated for bank 1 to bank 4
for j=1799:1802
    config_factory.NumChPopulated(j-1799+1,1) =
fread(logfile,1,'int16','b');
end

config_factory.SignalPolarity = fread(logfile,1,'int16','b'); %
1803: Signal polarity nibble based, 0=sign magnitude, 1=magnitude
config_factory.TestVoltageEnable = fread(logfile,1,'int16','b'); %
1804: Test voltage enable for bank 1 to bank 4

% 1805-1808: High voltage supply normalization values.
% Factory calculated values. Floating point calculation results * 100
for j=1805:1806
    for k=1:2
        config_factory.HVParameter(j-1805+1,1) =
fread(logfile,1,'int16','b');
    end
end

```

```

end
end

config_factory.AssemblyRevisionPCRev = fread(logfile,1,'int16','b');
% 1809: PCB Revision Number
config_factory.AssemblyRevisionLetter =
fread(logfile,1,'int16','b');% 1810: Assembly PC Letter
config_factory.Reserved1 = fread(logfile,1,'int16','b');
% 1811: Reserved
config_factory.X1 = fread(logfile,1,'int16','b');
% 1812: Trigger LED On Period
config_factory.X2 = fread(logfile,1,'int16','b');
% 1813: Trigger LED Off Period
config_factory.Y1 = fread(logfile,1,'int16','b');
% 1814: Acquisition LED Off Period
config_factory.Y2 = fread(logfile,1,'int16','b');
% 1815: Acquisition LED Off Period
config_factory.CPLDRevCode = fread(logfile,1,'int16','b');
% 1816: CPLD revision code
config_factory.ModelNumber = fscanf(logfile,'%c',32);
% 1817-1832: Model number
config_factory.SDRAMPopulated = fread(logfile,1,'int16','b');
% 1833: SDRAM Populated, 0=none, 1=32MByte, 2=64MByte
config_factory.SDRAMEnabled = fread(logfile,1,'int16','b');
% 1834: SDRAM Populated, 0=none, 1=32MByte, 2=64MByte
config_factory.Reserved2 = fread(logfile,1,'int16','b');
% 1835: Reserved

% 1836-1843: Bank 1 floating point programmable bit scale factor,
units of coulombs
for j=0:3
    config_factory.ProgScaling(j+1) = fread(logfile,1,'float','b');
end

fseek(logfile,(2000-1844)*2,0);
loc(3) = ftell(logfile);

%% Data Packets
clearvars data % Reinitializing
data % Go to beginning
fseek(logfile,data_offset*word_len,'bof');
of data block section
loc = ftell(logfile);

numChan = 32; % Number of
channels in our spectrometer
%L = 1 + numChan + 2; % Length of event
packet

F=0;
if config_user.DataFormat==0
    F=1;
end

K1 = round((config_user.NumChannelsB0+7)/8);
K2 = round((config_user.NumChannelsB1+7)/8);
K3 = round((config_user.NumChannelsB2+7)/8);
K4 = round((config_user.NumChannelsB3+7)/8);
L = 1 ...
    + (config_user.NumChannelsB0 + config_user.NumChannelsB1 +
config_user.NumChannelsB2 + config_user.NumChannelsB3) ...
    + (K1 + K2 + K3 + K4)*(F+config_user.RangeErrorEnable) ...

```

```

+ 2 * config_user.TimestampEnable ... % Timestamp
+ 2 * config_user.BoxcarWidthEnable ... % Boxcar width
+ 0 ... % ADC
+ 0; % External words

% An attempt to remove null spaces in the model number
model = double(config_factory.ModelNumber); % Need to convert to
double to find 0s
model(model==0) = []; % Remove all 0s
model = native2unicode(native2unicode(model)); % Reconvert the data
to nativev numbers
if contains(model, '580')==1 % Check to see if the
model number is IQSP580
    LSBWeight = 0.05951; % Least significant
weight bit conversion from number to pC
elseif contains(model, '480')==1 % Checking the data
format
    if config_user.DataFormat~=1
        LSBWeight = 0.02380;
    else
        LSBWeight = 0.04760;
    end
else
    sprintf('Unknown conversion factor from signal to pC')
    return;
end

k=0;

num_events = (logfile_info.bytes - loc)/(2*L);
%wait = waitbar(k/num_events, sprintf('Reading data...\nImage:
%i/%i', k, num_events));
wait = waitbar(k/num_events, sprintf('Reading data...', k, num_events));

while logfile_info.bytes - loc > L

    %waitbar(k/num_events, wait, sprintf('Reading data...\nImage:
%i/%i', k, num_events));
    % data - columns 1:6 is header of event packet info
    %     columns 7:38 is data
    %     columns 39:40 is time stamp indicators
    %     column 41 is the timestamp in ms

    k=k+1; % event counter variable

    temp = dec2bin(fread(logfile, 1, 'uint16', 'b'), 16);
    data(k, 1) = str2num(temp(1:3)); % Packet type
    data(k, 2) = str2num(temp(4)); % Out of range fault
    data(k, 3) = str2num(temp(5)); % Input error fault
    data(k, 4) = str2num(temp(6:10)); % Reserved
    data(k, 5) = str2num(temp(11)); % Filter match
    data(k, 6) = str2num(temp(12:16)); % Packet type

    for j = 7:38
        data(k, j) = fread(logfile, 1, 'int16', 'b')*LSBWeight;
    end

    % Timestamp
    % Says that this is supposed to be a 2-word (32 bit) variable
with the
    % "most significant word" following the "least significant word"
in the

```

```

        % event packet.

        % 06/02/2020 - I'm truly flabbergasted by how ridiculous this is,
so
        % buckle in. The "most significant word" is a 16 bit number that
gets
        % added to the value of (least significant word *
TimestampInterval).
        % What an amazingly ridiculous idea for formatting time. I'm not
sure
        % who had this idea or decided it was a good implementation, but
do you
        % have any idea how int it took me to figure that out? So long,
man.
        data(k,39) = fread(logfile,1,'uint16','b'); % most significant
word, added below
        data(k,40) = fread(logfile,1,'uint16','b'); % least significant
word, multiplied by timestamp interval
        data(k,41) =
(data(k,39)+((config_user.TimestampInterval/10)*(data(k,40))))*1e-7;
        % ms

        loc = ftell(logfile);
        if mod(k,10000)==0
            waitbar(k/num_events)
        end

    end
    close(wait)
    toc

%% Data plotting check
% figure
% hold on
% for k=1000:1000:250000
% plot3([1:32],ones(1,32)*(data(k,41)-data(1,41)),data(k,6:37))
% end
% view([25 25])
% xlabel('Channel')
% ylabel('Event #')

%% Close File
fclose(logfile)

end

```



## Bibliography

- <sup>1</sup> S.R. Turns, *An Introduction to Combustion: Concepts and Applications*, 3rd ed. (McGraw-Hill, New York, NY, 2012).
- <sup>2</sup> United States, *Military Explosives*. (Headquarters, Dept. of the Army, Washington, D.C., 1989).
- <sup>3</sup> M. Denny and A. McFadzean, *Rocket Science* (Springer, 2019).
- <sup>4</sup> C.M. Bishop, *Explosive Science* (The Royal Institution, 2012).
- <sup>5</sup> G.P. Sutton and O. Biblarz, *Rocket Propulsion Elements*, 8th ed. (Wiley, 2011).
- <sup>6</sup> V.E. Zarko and A.A. Gromov, *Energetic Nanomaterials: Synthesis, Characterization, and Application* (Elsevier, 2016).
- <sup>7</sup> M. Shukla, V.M. Boddu, J.A. Steevens, D. Reddy, and J. Leszczynski, *Energetic Materials From Cradle to Grave*. (Springer Verlag, 2017).
- <sup>8</sup> W.M. Deen, *Analysis of Transport Phenomena* (2013).
- <sup>9</sup> S. Bastea, L.E. Fried, K.R. Glaesemann, W.M. Howard, P.C. Sovers, and P.A. Vitello, User's Manual, Lawrence Livermore Natl. Lab. Livermore, CA (2006).
- <sup>10</sup> C.E.A. NASA, Softw. Packag. Ver 2, (n.d.).
- <sup>11</sup> G.C. Egan and M.R. Zachariah, Combust. Flame **162**, 2959 (2015).
- <sup>12</sup> M.R. Weismiller, J.Y. Malchi, J.G. Lee, R.A. Yetter, and T.J. Foley, Proc. Combust. Inst. **33**, 1989 (2011).
- <sup>13</sup> W.P. Bassett and D.D. Dlott, in *AIP Conf. Proc.* (2017).
- <sup>14</sup> J.M. Peuker, P. Lynch, H. Krier, and N. Glumac, Propellants, Explos. Pyrotech. **38**, 577 (2013).
- <sup>15</sup> J.M. Densmore, M.M. Biss, K.L. McNesby, and B.E. Homan, Appl. Opt. **50**, 2659 (2011).
- <sup>16</sup> D.K. Kim and P.B. Sunderland, Fire Saf. J. **106**, 88 (2019).
- <sup>17</sup> M. Alemohammad, E.R. Wainwright, J.R. Stroud, T.P. Weihs, and M.A. Foster, (2020).
- <sup>18</sup> J. Kalman, D. Allen, N. Glumac, and H. Krier, J. Thermophys. Heat Transf. **29**, 74 (2015).
- <sup>19</sup> J. Kalman and T. Hedman, Propellants, Explos. Pyrotech. **41**, 793 (2016).
- <sup>20</sup> C. Wyatt, *Radiometric Calibration: Theory and Methods* (Elsevier, 2012).
- <sup>21</sup> D.D. Dlott, Mater. Sci. Technol. **22**, 463 (2006).
- <sup>22</sup> M.R. Zachariah, Propellants, Explos. Pyrotech. **38**, 7 (2013).
- <sup>23</sup> J. Akhavan, *Chemistry of Explosives*, 3rd ed. (Royal Society of Chemistry, 2011).
- <sup>24</sup> E. Lafontaine and M. Comet, *Nanothermites* (John Wiley & Sons, 2016).
- <sup>25</sup> C. Rossi, Propellants, Explos. Pyrotech. **39**, 323 (2014).
- <sup>26</sup> S.H. Fischer and M.C. Grubelich, (1998).
- <sup>27</sup> D. Sundaram, V. Yang, and R.A. Yetter, Prog. Energy Combust. Sci. **61**, 293 (2017).
- <sup>28</sup> M.H. Ervin, S.S. Bedair, C.R. Knick, H. Tsang, B. Isaacson, and N.W. Piekiel, J. Microelectromechanical Syst. **26**, 1408 (2017).
- <sup>29</sup> S. Elbasuney, M. Gaber Zaky, M. Radwan, and S.F. Mostafa, Appl. Surf. Sci. **419**, 328 (2017).
- <sup>30</sup> G. Lengellé, J. Duterque, and J.F. Trubert, Nato **16**, 27 (2002).

- <sup>31</sup> S. Fischer and M. Grubelich, in *32nd Jt. Propuls. Conf. Exhib.* (1996).
- <sup>32</sup> M.L. Pantoya and J.J. Granier, *Propellants, Explos. Pyrotech.* **30**, 53 (2005).
- <sup>33</sup> S.F. Son, B.W. Asay, T.J. Foley, R.A. Yetter, M.H. Wu, and G.A. Risha, *J. Propuls. Power* **23**, 715 (2007).
- <sup>34</sup> M.C. Rehwoldt, H. Wang, D.J. Kline, T. Wu, N. Eckman, P. Wang, N.R. Agrawal, and M.R. Zachariah, *Combust. Flame* **211**, 260 (2020).
- <sup>35</sup> H. Wang, J. Shen, D.J. Kline, N. Eckman, N.R. Agrawal, T. Wu, P. Wang, and M.R. Zachariah, *Adv. Mater.* **31**, 1806575 (2019).
- <sup>36</sup> J. Zapata, A. Nicollet, B. Julien, G. Lahiner, A. Esteve, and C. Rossi, *Combust. Flame* **205**, 389 (2019).
- <sup>37</sup> B.S. Bockmon, M.L. Pantoya, S.F. Son, B.W. Asay, and J.T. Mang, *J. Appl. Phys.* **98**, 1 (2005).
- <sup>38</sup> K.T. Sullivan, J.D. Kuntz, and A.E. Gash, *Propellants, Explos. Pyrotech.* **39**, 407 (2014).
- <sup>39</sup> R.J. Jacob, D.L. Ortiz-Montalvo, K.R. Overdeep, T.P. Weihs, and M.R. Zachariah, *J. Appl. Phys.* **121**, (2017).
- <sup>40</sup> S. Wang, Y. Yang, H. Yu, and D.D. Dlott, *Propellants, Explos. Pyrotech.* **30**, 148 (2005).
- <sup>41</sup> G.Q. Jian, S. Chowdhury, K. Sullivan, and M.R. Zachariah, *Combust. Flame* **160**, 432 (2013).
- <sup>42</sup> C.E. Aumann, G.L. Skofronick, and J.A. Martin, *J. Vac. Sci. Tech. B* **13**, 1178 (1995).
- <sup>43</sup> V. Baijot, D.-R.R. Mehdi, C. Rossi, and A. Estève, *Combust. Flame* **180**, 10 (2017).
- <sup>44</sup> B.W. Asay, S.F. Son, J.R. Busse, and D.M. Oschwald, *Propellants, Explos. Pyrotech.* **29**, 216 (2004).
- <sup>45</sup> V.I. Levitas, M.L. Pantoya, and S. Dean, *Combust. Flame* **161**, 1668 (2014).
- <sup>46</sup> J. Hübner, M. Klaumünzer, M. Comet, C. Martin, L. Vidal, M. Schäfer, C. Kryschi, and D. Spitzer, *Combust. Flame* **184**, 186 (2017).
- <sup>47</sup> K.T. Sullivan, N.W. Piekriel, C. Wu, S. Chowdhury, S.T. Kelly, T.C. Hufnagel, K. Fezzaa, and M.R. Zachariah, *Combust. Flame* **159**, 2 (2012).
- <sup>48</sup> K. Sullivan, G. Young, and M.R. Zachariah, *Combust. Flame* **156**, 302 (2009).
- <sup>49</sup> V. Sundaram, K. V. Logan, and R.F. Speyer, *J. Mater. Res.* **12**, 2657 (1997).
- <sup>50</sup> X. Zhou, D. Xu, Q. Zhang, J. Lu, and K. Zhang, *ACS Appl. Mater. Interfaces* **5**, 7641 (2013).
- <sup>51</sup> S.K. Adams, N.W. Piekriel, M.H. Ervin, and C.J. Morris, **233108**, 1 (1992).
- <sup>52</sup> C.D. Malec, N.H. Voelcker, J.G. Shapter, and A. V. Ellis, *Mater. Lett.* **64**, 2517 (2010).
- <sup>53</sup> B. Berger, *Propellants, Explos. Pyrotech.* **30**, 27 (2005).
- <sup>54</sup> W. Zhao, X. Wang, H. Wang, T. Wu, D.J. Kline, M. Rehwoldt, H. Ren, M.R. Zachariah, S. Holdren, M. Rehwoldt, H. Ren, and M.R. Zachariah, *Combust. Flame* **212**, 245 (2020).
- <sup>55</sup> M.C. Rehwoldt, Y. Yang, H. Wang, S. Holdren, and M.R. Zachariah, *J. Phys. Chem. C* **122**, 10792 (2018).
- <sup>56</sup> C. Rossi, (n.d.).
- <sup>57</sup> V. Tanguay, S. Goroshin, A.J. Higgins, and F. Zhang, *Combust. Sci. Technol.* **181**,

- 670 (2009).
- <sup>58</sup> P. Chakraborty and M.R. Zachariah, *Combust. Flame* **161**, 1408 (2014).
- <sup>59</sup> D.A. Firmansyah, K. Sullivan, K.-S. Lee, Y.H. Kim, R. Zahaf, M.R. Zachariah, and D. Lee, *J. Phys. Chem. C* **116**, 404 (2012).
- <sup>60</sup> L. Zhou, N. Piekiet, S. Chowdhury, and M.R. Zachariah, *J. Phys. Chem. C* **114**, 14269 (2010).
- <sup>61</sup> K. Sullivan and M.R. Zachariah, *J. Propuls. Power* **26**, 467 (2010).
- <sup>62</sup> M.R. Weismiller, J.G. Lee, and R.A. Yetter, *Proc. Combust. Inst.* **33**, 1933 (2011).
- <sup>63</sup> F. Wang, J. Cai, J. Yu, C. Li, and Z. Yang, *ChemElectroChem* (2017).
- <sup>64</sup> K. Murakami, D. Mabuchi, I. Ueda, E. Study, B. Desai, and P. Mokashi, (2017).
- <sup>65</sup> V.E. Sanders, B.W. Asay, T.J. Foley, B.C. Tappan, A.N. Pacheco, and S.F. Son, *J. Propuls. Power* **23**, 707 (2007).
- <sup>66</sup> S.M. Umbrajkar, M. Schoenitz, and E.L. Dreizin, *Thermochim. Acta* **451**, 34 (2006).
- <sup>67</sup> K. Moore, M.L. Pantoya, and S.F. Son, *J. Propuls. Power* **23**, 181 (2007).
- <sup>68</sup> J.A. Puszyński, C.J. Bulian, and J.J. Swiatkiewicz, *J. Propuls. Power* **23**, 698 (2007).
- <sup>69</sup> X. Wang, W. Zhou, J.B. DeLisio, G.C. Egan, and M.R. Zachariah, *Phys. Chem. Chem. Phys.* **19**, 12749 (2017).
- <sup>70</sup> J.J. Granier and M.L. Pantoya, *Combust. Flame* **138**, 373 (2004).
- <sup>71</sup> M.L. Pantoya and J.J. Granier, *J. Therm. Anal. Calorim.* **85**, 37 (2006).
- <sup>72</sup> M. Schoenitz, S.M. Umbrajkar, and E.L. Dreizin, *J. Propuls. Power* **23**, 683 (2007).
- <sup>73</sup> R.J. Jacob, D.J. Kline, and M.R. Zachariah, *J. Appl. Phys.* **123**, 115902 (2018).
- <sup>74</sup> T. Wu, X. Wang, P.Y. Zavalij, J.B. DeLisio, H. Wang, and M.R. Zachariah, *Combust. Flame* **191**, 335 (2018).
- <sup>75</sup> R. Russell, S. Bless, and M. Pantoya, *J. Energ. Mater.* **29**, 175 (2011).
- <sup>76</sup> T.J. Fleck, A.K. Murray, I.E. Gunduz, S.F. Son, G.T.-C. Chiu, and J.F. Rhoads, *Addit. Manuf.* **17**, 176 (2017).
- <sup>77</sup> H. Wang, M. Rehwoldt, D.J. Kline, T. Wu, P. Wang, and M.R. Zachariah, *Combust. Flame* **201**, 181 (2019).
- <sup>78</sup> L.J. Groven and M.J. Mezger, in *Energ. Mater.* (CRC Press, 2017), pp. 115–128.
- <sup>79</sup> L. Zhou, N. Piekiet, S. Chowdhury, and M.R. Zachariah, *Rapid Commun. Mass Spectrom.* **23**, 194 (2009).
- <sup>80</sup> X. Wang, T. Wu, and M.R. Zachariah, *J. Phys. Chem. C* **121**, 147 (2017).
- <sup>81</sup> S. Chowdhury, K. Sullivan, N. Piekiet, L. Zhou, and M.R. Zachariah, *J. Phys. Chem. C* **114**, 9191 (2010).
- <sup>82</sup> D.W. Hammerstroem, M.A. Burgers, S.W. Chung, E.A. Gulians, C.E. Bunker, K.M. Wentz, S.E. Hayes, S.W. Buckner, and P.A. Jelliss, *Inorg. Chem.* **50**, 5054 (2011).
- <sup>83</sup> J. Wang, Z. Qiao, Y. Yang, J. Shen, Z. Long, Z. Li, X. Cui, and G. Yang, *Chem. - A Eur. J.* **22**, 279 (2016).
- <sup>84</sup> D.A. Kaplowitz, G. Jian, K. Gaskell, R. Jacob, and M.R. Zachariah, *J. Energ. Mater.* **32**, 94 (2014).
- <sup>85</sup> Y. Chen, G.C. Egan, J. Wan, S. Zhu, R.J. Jacob, W. Zhou, J. Dai, Y. Wang, V.A. Danner, Y. Yao, K. Fu, Y. Wang, W. Bao, T. Li, M.R. Zachariah, and L. Hu, *Nat. Commun.* **7**, 12332 (2016).

- <sup>86</sup> H. Wang, D.J. Kline, M. Rehwoldt, T. Wu, W. Zhao, X. Wang, and M.R. Zachariah, *ACS Appl. Polym. Mater.* **1**, 982 (2019).
- <sup>87</sup> K.T. Sullivan, C. Zhu, E.B. Duoss, A.E. Gash, D.B. Kolesky, J.D. Kuntz, J.A. Lewis, and C.M. Spadaccini, *Adv. Mater.* **28**, 1934 (2016).
- <sup>88</sup> M.S. McClain, A. Afriat, J.F. Rhoads, I.E. Gunduz, and S.F. Son, *Propellants, Explos. Pyrotech.* **1** (2020).
- <sup>89</sup> E.R. Wainwright, K.T. Sullivan, and M.D. Grapes, *Adv. Eng. Mater.* **1901196**, 1 (2019).
- <sup>90</sup> A.M. Golobic, M.D. Durban, S.E. Fisher, M.D. Grapes, J.M. Ortega, C.M. Spadaccini, E.B. Duoss, A.E. Gash, and K.T. Sullivan, *Adv. Eng. Mater.* **21**, 1900147 (2019).
- <sup>91</sup> D.K. Smith, D.K. Unruh, C.C. Wu, and M.L. Pantoya, *J. Phys. Chem. C* **121**, 23184 (2017).
- <sup>92</sup> J. Shen, Z. Qiao, J. Wang, G. Yang, J. Chen, Z. Li, X. Liao, H. Wang, and M.R. Zachariah, *Thermochim. Acta* **666**, 60 (2018).
- <sup>93</sup> J. Dai, F. Wang, C. Ru, J. Xu, C. Wang, W. Zhang, Y. Ye, and R. Shen, *J. Phys. Chem. C* (2018).
- <sup>94</sup> L.H. Caveny and R.L. Glick, *J. Spacecr. Rockets* **4**, 79 (1967).
- <sup>95</sup> H. Wang, J.B. DeLisio, S. Holdren, T. Wu, Y. Yang, J. Hu, and M.R. Zachariah, *Adv. Eng. Mater.* **20**, 1 (2018).
- <sup>96</sup> H.S. Kim, J.H. Kim, K.J. Kim, and S.H. Kim, *Combust. Flame* **194**, 264 (2018).
- <sup>97</sup> B. Julien, J. Cure, L. Salvagnac, C. Josse, A. Esteve, and C. Rossi, *ACS Appl. Nano Mater.* (2020).
- <sup>98</sup> C.E. Aumann, *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.* **13**, 1178 (1995).
- <sup>99</sup> R.J. Jacob, G. Jian, P.M. Guerieri, and M.R. Zachariah, *Combust. Flame* **162**, 258 (2015).
- <sup>100</sup> H. Wang, R.J. Jacob, J.B. DeLisio, and M.R. Zachariah, *Combust. Flame* **180**, 175 (2017).
- <sup>101</sup> S.B. Kim, K.J. Kim, M.H. Cho, J.H. Kim, K.T. Kim, and S.H. Kim, *ACS Appl. Mater. Interfaces* **8**, 9405 (2016).
- <sup>102</sup> X. Zhou, M. Torabi, J. Lu, R. Shen, and K. Zhang, *ACS Appl. Mater. Interfaces* **6**, 3058 (2014).
- <sup>103</sup> R.J. Jacob and M.R. Zachariah, (2016).
- <sup>104</sup> R. Thiruvengadathan, A. Bezmelnitsyn, S. Apperson, C. Staley, P. Redner, W. Balas, S. Nicolich, D. Kapoor, K. Gangopadhyay, and S. Gangopadhyay, *Combust. Flame* **158**, 964 (2011).
- <sup>105</sup> S. Zhang and E.L. Dreizin, *J. Phys. Chem. C* **117**, 14025 (2013).
- <sup>106</sup> K. Kappagantula, C. Crane, and M. Pantoya, *Propellants, Explos. Pyrotech.* **39**, 434 (2014).
- <sup>107</sup> W.L. Perry, B.C. Tappan, B.L. Reardon, V.E. Sanders, and S.F. Son, *J. Appl. Phys.* **101**, 064313 (2007).
- <sup>108</sup> D. Ng and G. Fralick, *Rev. Sci. Instrum.* **72**, 1522 (2001).
- <sup>109</sup> T. Fu, Z. Wang, and X. Cheng, *J. Heat Transfer* **132**, 051602 (2010).
- <sup>110</sup> D.P. DeWitt and G.D. Nutter, editors, *Theory and Practice of Radiation Thermometry* (John Wiley & Sons, Inc., Hoboken, NJ, USA, 1988).

- <sup>111</sup> S. Goroshin, D.L. Frost, J. Levine, A. Yoshinaka, and F. Zhang, *Propellants Explos. Pyrotech.* **31**, 169 (2006).
- <sup>112</sup> P. Lynch, G. Fiore, H. Krier, and N. Glumac, *Combust. Sci. Technol.* **182**, 842 (2010).
- <sup>113</sup> D.C. Morton, *Astrophys. J. Suppl. Ser.* **149**, 205 (2003).
- <sup>114</sup> G.C. Egan, T. LaGrange, and M.R. Zachariah, *J. Phys. Chem. C* **119**, 2792 (2015).
- <sup>115</sup> P. Bucher, R.A. Yetter, F.L. Dryer, T.P. Parr, and D.M. Hanson-Parr, *Twenty-Seventh Symp. Combust. Vols 1 2* 2421 (1998).
- <sup>116</sup> K.T. Sullivan, O. Cervantes, J.M. Densmore, J.D. Kuntz, A.E. Gash, and J.D. Molitoris, *Propellants, Explos. Pyrotech.* **40**, 394 (2015).
- <sup>117</sup> M. Hu, Z. Yao, and X. Wang, *Ind. Eng. Chem. Res.* **56**, 3477 (2017).
- <sup>118</sup> A. Eftekhari and P. Jafarkhani, *J. Phys. Chem. C* **117**, 25845 (2013).
- <sup>119</sup> K. Fu, Y. Wang, C. Yan, Y. Yao, Y. Chen, J. Dai, S. Lacey, Y. Wang, J. Wan, T. Li, Z. Wang, Y. Xu, and L. Hu, *Adv. Mater.* **28**, 2587 (2016).
- <sup>120</sup> J.H. Kim, W.S. Chang, D. Kim, J.R. Yang, J.T. Han, G.W. Lee, J.T. Kim, and S.K. Seol, *Adv. Mater.* **27**, 157 (2015).
- <sup>121</sup> A. Johansson, P. Koskinen, J. Aumanen, H. Tsai, C. Chen, L. Chang, V. Hiltunen, J. Manninen, W.Y. Woon, and M. Pettersson, *1* (2017).
- <sup>122</sup> Z. Gao, J. Yang, J. Huang, Q. Yang, and C. Xiong, *Nanoscale* (2017).
- <sup>123</sup> K. Kostarelos and K.S. Novoselov, *Science (80-. )*. **344**, 261 (2014).
- <sup>124</sup> A. Servant, A. Bianco, M. Prato, and K. Kostarelos, *Bioorganic Med. Chem. Lett.* **24**, 1638 (2014).
- <sup>125</sup> H. Liu, L. Zhang, M. Yan, and J. Yu, *J. Mater. Chem. B* **5**, 6437 (2017).
- <sup>126</sup> Y. Yao, Z. Huang, P. Xie, S.D. Lacey, R.J. Jacob, H. Xie, F. Chen, A. Nie, T. Pu, and M. Rehwoldt, *Science (80-. )*. **359**, 1489 (2018).
- <sup>127</sup> M. Khan, M.N. Tahir, S.F. Adil, H.U. Khan, M.R.H. Siddiqui, A.A. Al-warthan, and W. Tremel, *J. Mater. Chem. A* **3**, 18753 (2015).
- <sup>128</sup> M.J. Allen, V.C. Tung, and R.B. Kaner, *Chem. Rev.* **110**, 132 (2010).
- <sup>129</sup> S. Eigler, C. Dotzer, and A. Hirsch, *Carbon N. Y.* **50**, 3666 (2012).
- <sup>130</sup> Y. Qiu, F. Guo, R. Hurt, and I. Külaots, *Carbon N. Y.* **72**, 215 (2014).
- <sup>131</sup> Y. Yao, F. Chen, A. Nie, S.D. Lacey, R.J. Jacob, S. Xu, Z. Huang, K. Fu, J. Dai, L. Salamanca-Riba, M.R. Zachariah, R. Shahbazian-Yassar, and L. Hu, *ACS Cent. Sci.* **3**, 294 (2017).
- <sup>132</sup> Y. Yang, M. Romano, G. Feng, X. Wang, T. Wu, S. Holdren, and M.R. Zachariah, *Langmuir* **34**, 585 (2018).
- <sup>133</sup> J.M. Densmore, B.E. Homan, M.M. Biss, and K.L. McNesby, *Appl. Opt.* **50**, 6267 (2011).
- <sup>134</sup> E.L. Dreizin, *Prog. Energy Combust. Sci.* **35**, 141 (2009).
- <sup>135</sup> M.W. Beckstead, *J. Chem. Inf. Model.* **53**, 1689 (2013).
- <sup>136</sup> Vision Research, (n.d.).
- <sup>137</sup> W. Rasband, U. S. Natl. Institutes Heal. Bethesda, Maryland, USA [//imagej.nih.gov/ij/](http://imagej.nih.gov/ij/) (2012).
- <sup>138</sup> N.M. Nguyen Huynh, Z.A. Boeva, J.-H. Smått, M. Pesonen, and T. Lindfors, *RSC Adv.* **8**, 17645 (2018).
- <sup>139</sup> H. Yamaguchi, J. Granstrom, W. Nie, H. Sojoudi, T. Fujita, D. Voiry, M. Chen, G. Gupta, A.D. Mohite, S. Graham, and M. Chhowalla, *Adv. Energy Mater.* n/a (2013).

- <sup>140</sup> Y. Su, V.G. Kravets, S.L. Wong, J. Waters, A.K. Geim, and R.R. Nair, *Nat. Commun.* **5**, 4843 (2014).
- <sup>141</sup> D. Kang, J.Y. Kwon, H. Cho, J.H. Sim, H.S. Hwang, C.S. Kim, Y.J. Kim, R.S. Ruoff, and H.S. Shin, *ACS Nano* **6**, 7763 (2012).
- <sup>142</sup> J.D. Renteria, S. Ramirez, H. Malekpour, B. Alonso, A. Centeno, A. Zurutuza, A.I. Cocemasov, D.L. Nika, and A.A. Balandin, *Adv. Funct. Mater.* **25**, 4664 (2015).
- <sup>143</sup> D. Ma, X. Li, Y. Guo, and Y. Zeng, (2018).
- <sup>144</sup> D.L. Hastings, M. Schoenitz, and E.L. Dreizin, *J. Alloys Compd.* **735**, 1863 (2018).
- <sup>145</sup> D.J. Shin, W.D. Kim, S. Lee, and D.C. Lee, *Combust. Flame* (2018).
- <sup>146</sup> T. Calais, A. Bancaud, A. Estève, and C. Rossi, *ACS Appl. Nano Mater.* **1**, 4716 (2018).
- <sup>147</sup> W. He, P.-J. Liu, G.-Q. He, M. Gozin, and Q.-L. Yan, *Adv. Mater.* **30**, 1706293 (2018).
- <sup>148</sup> K. Meeks, D.K. Smith, B. Clark, and M.L. Pantoya, *J. Mater. Chem. A* **5**, 7200 (2017).
- <sup>149</sup> M.R. Weismiller, Z.J. Huba, S.G. Tuttle, A. Epshteyn, and B.T. Fisher, *Combust. Flame* **176**, 361 (2017).
- <sup>150</sup> J.T. Koenig, A.P. Shaw, J.C. Poret, W.S. Eck, and L.J. Groven, *ACS Sustain. Chem. Eng.* **5**, 9477 (2017).
- <sup>151</sup> N. Zohari, M.H. Keshavarz, and S.A. Seyedsadjadi, *Cent. Eur. J. Energ. Mater.* **10**, (2013).
- <sup>152</sup> Y. Zong, R.J. Jacob, S. Li, and M.R. Zachariah, *J. Phys. Chem. A* **119**, 6171 (2015).
- <sup>153</sup> G.C. Egan, K.T. Sullivan, T. LaGrange, B.W. Reed, and M.R. Zachariah, *J. Appl. Phys.* **115**, 84903 (2014).
- <sup>154</sup> K.T. Sullivan, W.-A. Chiou, R. Fiore, and M.R. Zachariah, *Appl. Phys. Lett.* **97**, 133104 (2010).
- <sup>155</sup> J.B. DeLisio, G.C. Egan, S.-C. Liou, W.-A. Chiou, and M.R. Zachariah, *Microsc. Microanal.* **22**, 780 (2016).
- <sup>156</sup> B.J. Henz, T. Hawa, and M.R. Zachariah, *J. Appl. Phys.* **107**, 24901 (2010).
- <sup>157</sup> G.D. Scott and D.M. Kilgour, *J. Phys. D. Appl. Phys.* **2**, 863 (1969).
- <sup>158</sup> C.D. Zangmeister, J.G. Radney, L.T. Dockery, J.T. Young, X. Ma, R. You, and M.R. Zachariah, *Proc. Natl. Acad. Sci.* **111**, 9037 LP (2014).
- <sup>159</sup> H. Wang, G. Jian, G.C. Egan, and M.R. Zachariah, *Combust. Flame* **161**, 2203 (2014).
- <sup>160</sup> M. Comet, C. Martin, F. Schnell, and D. Spitzer, *Propellants, Explos. Pyrotech.* **44**, 18 (2018).
- <sup>161</sup> H. Wang, D.J. Kline, M. Rehwoldt, and M.R. Zachariah, *Propellants, Explos. Pyrotech.* **43**, 977 (2018).
- <sup>162</sup> S. Wang, M. Schoenitz, S.A. Grinshpun, M. Yermakov, and E.L. Dreizin, *J. Aerosol Sci.* **116**, 106 (2018).
- <sup>163</sup> J.L. Gottfried, D.K. Smith, C.C. Wu, and M.L. Pantoya, *Sci. Rep.* **8**, 1 (2018).
- <sup>164</sup> H. Wang, S. Holdren, and M.R. Zachariah, *Combust. Flame* **197**, 120 (2018).
- <sup>165</sup> M.A. Hobosyan and K.S. Martirosyan, *Propellants, Explos. Pyrotech.* **42**, 506 (2017).

- <sup>166</sup> H. Wang, D.J. Kline, and M.R. Zachariah, *Nat. Commun.* **10**, 3032 (2019).
- <sup>167</sup> MathWorks, (2019).
- <sup>168</sup> G. Jian, S. Chowdhury, J. Feng, and M.R. Zachariah, in *8th US Natl. Combust. Meet. 2013* (2013).
- <sup>169</sup> D.K. Smith, M.L. Pantoya, J.S. Parkey, and M. Kesmez, *RSC Adv.* **7**, 10183 (2017).
- <sup>170</sup> W. Zhou, J. Zuo, and W. Ren, *Compos. Part A Appl. Sci. Manuf.* **43**, 658 (2012).
- <sup>171</sup> W.P. Bassett and D.D. Dlott, *Rev. Sci. Instrum.* **87**, (2016).
- <sup>172</sup> D.J. Kline, M.C. Rehwoldt, C.J. Turner, P. Biswas, G.W. Mulholland, S.M. McDonnell, and M.R. Zachariah, *J. Appl. Phys.* **127**, 55901 (2020).
- <sup>173</sup> R.J. Jacob, K.J. Hill, Y. Yang, M.L. Pantoya, and M.R. Zachariah, *Combust. Flame* **205**, 33 (2019).
- <sup>174</sup> D.J. Kline, M.C. Rehwoldt, H. Wang, N.E. Eckman, and M.R. Zachariah, *Appl. Phys. Lett.* **115**, 114101 (2019).
- <sup>175</sup> J. Shen, H. Wang, D.J. Kline, Y. Yang, X. Wang, M. Rehwoldt, T. Wu, S. Holdren, and M.R. Zachariah, *Combust. Flame* **215**, 86 (2020).
- <sup>176</sup> S. Isert, C.D. Lane, I.E. Gunduz, and S.F. Son, *Proc. Combust. Inst.* **36**, 2283 (2017).
- <sup>177</sup> A. Varma, A.S. Rogachev, A.S. Mukasyan, and S. Hwang, *Proc. Natl. Acad. Sci. U. S. A.* **95**, 11053 (1998).
- <sup>178</sup> I.P. Borovinskaya, A. Gromov, E.A. Levashov, Y.M. Maksimov, A.S. Mukasyan, and A.S. Rogachev, *Concise Encyclopedia of Self-Propagating High-Temperature Synthesis : History, Theory, Technology, and Products* (Elsevier, 2017).
- <sup>179</sup> A.S. Mukasyan and C.E. Shuck, *Int. J. Self-Propagating High-Temperature Synth.* **26**, 145 (2017).
- <sup>180</sup> M.S. Song, M.W. Ran, and Y.Y. Kong, *Int. J. Refract. Met. Hard Mater.* **29**, 392 (2011).
- <sup>181</sup> A.S. Mukasyan, A.S. Rogachev, M. Mercedes, and A. Varma, *Chem. Eng. Sci.* **59**, 5099 (2004).
- <sup>182</sup> A.S. Mukasyan, A.S. Rogachev, and A. Varma, *Chem. Eng. Sci.* **54**, 3357 (1999).
- <sup>183</sup> S. Vummidi Lakshman, J.D. Gibbins, E.R. Wainwright, and T.P. Weihs, *Powder Technol.* **343**, 87 (2019).
- <sup>184</sup> M. Schoenitz, T.S. Ward, and E.L. Dreizin, *Proc. Combust. Inst.* **30**, 2071 (2005).
- <sup>185</sup> S. Hwang, A.S. Mukasyan, and A. Varma, *Combust. Flame* **115**, 354 (1998).
- <sup>186</sup> A.G. Merzhanov, A.S. Mukas'yan, A.S. Rogachev, A.E. Sychev, S. Hwang, and A. Varma, *Combust. Explos. Shock Waves* **32**, 655 (1996).
- <sup>187</sup> F.D. Tang, A.J. Higgins, and S. Goroshin, *Combust. Theory Model.* **13**, 319 (2009).
- <sup>188</sup> S.Q. Arlington, S. Vummidi Lakshman, S.C. Barron, J.B. DeLisio, J.C. Rodriguez, S. Narayanan, G.M. Fritz, and T.P. Weihs, *Mater. Adv.* **1**, 1151 (2020).
- <sup>189</sup> S.Q. Arlington, S.C. Barron, J.B. DeLisio, J.C. Rodriguez, S. Vummidi Lakshman, and T.P. Weihs, (2020).
- <sup>190</sup> G.M. Fritz and S. Barron, (2019).
- <sup>191</sup> M. Song, B. Huang, M. Zhang, and J. Li, *ISIJ Int.* **48**, 1026 (2008).
- <sup>192</sup> M.S. Song, B. Huang, M.X. Zhang, and J.G. Li, *Powder Technol.* **191**, 34 (2009).
- <sup>193</sup> Q. Hu, M. Zhang, P. Luo, M. Song, and J. Li, *Int. J. Refract. Met. Hard Mater.* **35**,

- 251 (2012).
- <sup>194</sup> S.G. Vadchenko, D.Y. Kovalev, and M.A. Luginina, *Combust. Explos. Shock Waves* **53**, 171 (2017).
- <sup>195</sup> T. Boddington, P.G. Laye, J. Tipping, and D. Whalley, *Combust. Flame* **63**, 359 (1986).
- <sup>196</sup> S.D. Dunmead, Z.A. Munir, and J.B. Holt, *J. Am. Ceram. Soc.* **75**, 180 (1992).
- <sup>197</sup> A. Khawam and D.R. Flanagan, *J. Phys. Chem. B* **110**, 17315 (2006).
- <sup>198</sup> F. Baras, *J. Alloys Compd.* **455**, 113 (2008).
- <sup>199</sup> M. Marinšek, J. Kemperl, B. Likozar, and J. Maček, *Ind. Eng. Chem. Res.* **47**, 4379 (2008).
- <sup>200</sup> A.G. Merzhanov, *Dokl. Akad. Nauk SSSR* **233**, 1130 (1977).
- <sup>201</sup> G. Reeves, (2009).
- <sup>202</sup> A. Varma, A.S. Mukasyan, and S. Hwang, *Chem. Eng. Sci.* **56**, 1459 (2001).
- <sup>203</sup> M. Alatalo, M. Weinert, and R.E. Watson, *Phys. Rev. B* **57**, R2009 (1998).
- <sup>204</sup> J.M. Beck and V.A. Volpert, *Phys. D Nonlinear Phenom.* **182**, 86 (2003).
- <sup>205</sup> A.S. Rogachev, *Combust. Explos. Shock Waves* **39**, 150 (2003).
- <sup>206</sup> A.S. Rogachev and F. Baras, *Int. J. Self-Propagating High-Temperature Synth.* **16**, 141 (2007).
- <sup>207</sup> J.Y. Zhang, Z.Y. Fu, W.M. Wang, and Q.J. Zhang, *J. Wuhan Univ. Technol. Mater. Sci. Ed.* **18**, 32 (2003).
- <sup>208</sup> K.G. Shkadinskii, B.I. Khaikin, and A.G. Merzhanov, *Combust. Explos. Shock Waves* **7**, 15 (1971).
- <sup>209</sup> M.G. Lakshmikantha and J.A. Sekhar, *Metall. Trans. A* **24**, 617 (1993).
- <sup>210</sup> A. Bayliss, A. Bayliss, B.J. Matkowsky, and B.J. Matkowsky, *SIAM J. Appl. Math.* **50**, 437 (1990).
- <sup>211</sup> A.S. Mukasyan, A.S. Rogachev, and A. Varma, *AIChE J.* **45**, 2580 (1999).
- <sup>212</sup> K.G. Shkadinsky and A.S. Rogachev, in edited by I.P. Borovinskaya, A.A. Gromov, E.A. Levashov, Y.M. Maksimov, A.S. Mukasyan, and A.S.B.T.-C.E. of S.-P.H.-T.S. Rogachev (Elsevier, Amsterdam, 2017), pp. 22–23.
- <sup>213</sup> P.M. Guerieri, S. DeCarlo, B. Eichhorn, T. Connell, R.A. Yetter, X. Tang, Z. Hicks, K.H. Bowen, and M.R. Zachariah, *J. Phys. Chem. A* **119**, 11084 (2015).
- <sup>214</sup> C. Rossi, *Propellants, Explos. Pyrotech.* **1** (2018).
- <sup>215</sup> I.E. Gunduz, M.S. McClain, P. Cattani, G.T.C. Chiu, J.F. Rhoads, and S.F. Son, *Addit. Manuf.* **22**, 98 (2018).
- <sup>216</sup> N. Kubota, M. Ichidat, and T. Fujisawa, *AIAA J.* **20**, 116 (1982).
- <sup>217</sup> C. Shuling and L. Fengsheng, *Combust. Flame* **45**, 213 (1982).
- <sup>218</sup> S. Shioya, M. Kohga, and T. Naya, *Combust. Flame* **161**, 620 (2014).
- <sup>219</sup> H. Wang, D.J. Kline, and M.R. Zachariah, *Nat. Commun.* **1** (2019).
- <sup>220</sup> H. Nagano, H. Kato, A. Ohnishi, and Y. Nagasaka, *High Temp. - High Press.* **33**, 253 (2001).
- <sup>221</sup> T. Katsura, *Phys. Chem. Miner.* **20**, 201 (1993).
- <sup>222</sup> J. Yu, R. Qian, and P. Jiang, *Fibers Polym.* **14**, 1317 (2013).
- <sup>223</sup> F.G.S. Filler, J. Yu, R. Qian, and P. Jiang, (2013).
- <sup>224</sup> S.C. Cheng and R.I. Vachon, *Int. J. Heat Mass Transf.* **12**, 249 (1969).
- <sup>225</sup> S.N. Maiti and K. Ghosh, *J. Appl. Polym. Sci.* **52**, 1091 (1994).
- <sup>226</sup> F. Kargar, Z. Barani, R. Salgado, B. Debnath, J.S. Lewis, E. Aytaç, R.K. Lake,



- and A.A. Balandin, ACS Appl. Mater. Interfaces **10**, 37555 (2018).
- <sup>227</sup> J. Yuan, S. Yao, W. Li, A. Sylvestre, and J. Bai, J. Phys. Chem. C **121**, 12063 (2017).
- <sup>228</sup> D.K. Smith, J. Cano, and M.L. Pantoya, J. Thermophys. Heat Transf. **31**, 1 (2017).
- <sup>229</sup> R. Pal, Compos. Part A Appl. Sci. Manuf. **39**, 718 (2008).
- <sup>230</sup> K. Pietrak and T. Wiśniewski, J. Power Technol. **95**, 14 (2015).
- <sup>231</sup> L.E. Nielsen, Ind. Eng. Chem. Fundam. **13**, 17 (1974).
- <sup>232</sup> Z. Dang, J. Yuan, S. Yao, and R. Liao, Adv. Mater. **25**, 6334 (2013).
- <sup>233</sup> D. Wanasinghe, F. Aslani, G. Ma, and D. Habibi, Nanomaterials **10**, 1 (2020).
- <sup>234</sup> C. Du, M. Li, M. Cao, S. Feng, and H. Guo, Carbon N. Y. **126**, 197 (2018).
- <sup>235</sup> Z.D. Lawless, M.L. Hobbs, and M.J. Kaneshige, J. Energ. Mater. **38**, 214 (2020).
- <sup>236</sup> S. Agarwal, M.M.K. Khan, and R.K. Gupta, Polym. Eng. Sci. **48**, 2475 (2008).
- <sup>237</sup> R. Zheng, J. Gao, J. Wang, S.P. Feng, H. Ohtani, J. Wang, and G. Chen, Nano Lett. **12**, 188 (2012).
- <sup>238</sup> D.J. Kline, Z. Alibay, M.C. Rehwoldt, A. Idrogo-Lam, S.G. Hamilton, P. Biswas, F. Xu, and M.R. Zachariah, Combust. Flame **215**, 417 (2020).
- <sup>239</sup> S. Yan, G. Jian, and M.R. Zachariah, ACS Appl. Mater. Interfaces **4**, 6432 (2012).
- <sup>240</sup> R. Chen, Y. Luo, J. Sun, and G. Li, Propellants, Explos. Pyrotech. **37**, 422 (2012).
- <sup>241</sup> M.S. McClain, I.E. Gunduz, and S.F. Son, Proc. Combust. Inst. **37**, 3135 (2019).
- <sup>242</sup> H. Wang, J.B. Delisio, S. Holdren, T. Wu, Y. Yang, J. Hu, and M.R. Zachariah, Adv. Eng. Mater. **20**, 1700547 (2018).
- <sup>243</sup> T. An, W. Qu, Y. Yang, F. Zhao, and Q. Yan, *Preparation, Characterization, and Application of Superthermites in Solid Propellants* (Elsevier Inc., 2019).
- <sup>244</sup> C. Huang, G. Jian, J.B. De Lisio, H. Wang, and M.R. Zachariah, Adv. Eng. Mater. **17**, 95 (2015).
- <sup>245</sup> N. V Muravyev, K.A. Monogarov, U. Schaller, I. V Fomenkov, and A.N. Pivkina, Propellants, Explos. Pyrotech. **44**, 941 (2019).
- <sup>246</sup> S. Goroshin, J.H.S. Lee, and Y. Shoshin, in *Twenty-Seventh Symp. Combust. Combust. Inst.* (1998), pp. 743–749.
- <sup>247</sup> A. Bacciochini, M.I. Radulescu, M. Yandouzi, G. Maines, J.J. Lee, and B. Jodoin, Surf. Coatings Technol. **226**, 60 (2013).
- <sup>248</sup> V.J. Ervin, J.W. Klett, and C.M. Mundt, J. Mater. Sci. **34**, 3545 (1999).
- <sup>249</sup> A.G. Merzhanov, Combust. Flame **13**, 143 (1969).
- <sup>250</sup> A.S. Mukasyan and A.S. Rogachev, Prog. Energy Combust. Sci. **34**, 377 (2008).
- <sup>251</sup> V.K. Smolyakov, Combust. Explos. Shock Waves **37**, 274 (2001).
- <sup>252</sup> N. Chakraborty and N. Swaminathan, Phys. Fluids **19**, (2007).
- <sup>253</sup> M. Oberlack, R. Arlitt, and N. Peter, Combust. Theory Model. **4**, 495 (2000).
- <sup>254</sup> P.S. Grinchuk, J. Eng. Phys. Thermophys. **86**, 875 (2013).
- <sup>255</sup> Becker and Kaus, in *Numer. Model. Earth Syst.* (2016), pp. 1–9.
- <sup>256</sup> L. Niragire, 2D Numerical Heat Inverse Model for Diffusivity Coefficients in Cartesian Coordinates, 2013.
- <sup>257</sup> G.K. Bama, P.I. Devi, and K. Ramachandran, J. Mater. Res. **44**, 1302 (2009).
- <sup>258</sup> G. Lahiner, A. Nicollet, J. Zapata, L. Marín, N. Richard, M.D. Rouhani, C. Rossi, and A. Estève, J. Appl. Phys. **122**, 155105 (2017).
- <sup>259</sup> H.P. Langtangen and S. Linge, *Finite Difference Methods for Diffusion Processes* (2016).

- <sup>260</sup> G.W. Recktenwald, *Finite-Difference Approximations to the Heat Equation* (2011).
- <sup>261</sup> P. Wesseling, IMA J. Numer. Anal. **583** (1996).
- <sup>262</sup> J.B. DeLisio, X. Wang, T. Wu, G.C. Egan, R.J. Jacob, and M.R. Zachariah, J. Appl. Phys. **122**, 245901 (2017).
- <sup>263</sup> Y. Chen, D.R. Guildenbecher, K.N.G. Hoffmeister, M.A. Cooper, H.L. Stauffacher, M.S. Oliver, and E.B. Washburn, Combust. Flame **182**, 225 (2017).
- <sup>264</sup> S.C. Ligon, R. Liska, J. Stampfl, M. Gurr, and R. Mülhaupt, Chem. Rev. **117**, 10212 (2017).
- <sup>265</sup> S.J. Barkley, K. Zhu, J.B. Michael, and T.R. Sippel, in *AIAA Scitech 2019 Forum* (American Institute of Aeronautics and Astronautics, 2019).
- <sup>266</sup> T. Sippel, *Pulsed Microwave Plasma Instrumentation for Investigation of Plasma-Tuned Multiphase Combustion Final Report DISTRIBUTION A: Distribution Approved for Public Release* (n.d.).
- <sup>267</sup> J.D. Katz, Annu. Rev. Mater. Sci. **22**, 153 (1992).
- <sup>268</sup> C. Leonelli, P. Veronesi, L. Denti, A. Gatto, and L. Iuliano, J. Mater. Process. Technol. **205**, 489 (2008).
- <sup>269</sup> R. Roy, D. Agrawal, J. Cheng, and S. Gedevarishvili, Nature **399**, 668 (1999).
- <sup>270</sup> G.M. Batanov, N.K. Bereghetskaya, V.A. Kopiev, I.A. Kossyi, A.N. Magunov, V.A. Shcherbakov, and N. V Sachkova, Dokl. Phys. **51**, 180 (2006).
- <sup>271</sup> S. Horikoshi and N. Serpone, *Microwaves in Nanoparticle Synthesis: Fundamentals and Applications* (John Wiley & Sons, 2013).
- <sup>272</sup> K. Hasue, M. Tanabe, N. Watanabe, S. Nakahara, F. Okada, and A. Iwama, Propellants, Explos. Pyrotech. **15**, 181 (1990).
- <sup>273</sup> S. Horikoshi, R.F. Schiffmann, J. Fukushima, and N. Serpone, *Microwave Chemical and Materials Processing* (Springer, 2018).
- <sup>274</sup> R. Kaur, M. Newborough, and S.D. Probert, Appl. Energy **44**, 337 (1993).
- <sup>275</sup> C.A. Crane, M.L. Pantoya, B.L. Weeks, and M. Saed, Powder Technol. **256**, 113 (2014).
- <sup>276</sup> M. Ignatenko, M. Tanaka, and M. Sato, Jpn. J. Appl. Phys. **48**, (2009).
- <sup>277</sup> A. Porch, D. Slocombe, and P.P. Edwards, Phys. Chem. Chem. Phys. **15**, 2757 (2013).
- <sup>278</sup> N. Yoshikawa, E. Ishizuka, and S. Taniguchi, Mater. Trans. **47**, 898 (2006).
- <sup>279</sup> Y. Meir and E. Jerby, Combust. Flame **159**, 2474 (2012).
- <sup>280</sup> C.A. Crane, M.L. Pantoya, and B.L. Weeks, J. Appl. Phys. **115**, 104106 (2014).
- <sup>281</sup> T. Kim, J. Lee, and K.H. Lee, Carbon Lett. **15**, 15 (2014).
- <sup>282</sup> S. Chandrasekaran, T. Basak, and R. Srinivasan, Int. Commun. Heat Mass Transf. **48**, 22 (2013).
- <sup>283</sup> C.F. Bohren and D.R. Huffman, *Absorption and Scattering of Light by Small Particles* (John Wiley & Sons, 2008).
- <sup>284</sup> P. Biswas, G. Mulholland, M. Rehwoldt, D.J. Kline, and M.R. Zachariah, J. Quant. Spectrosc. Radiat. Transf. **247**, 106938 (2020).
- <sup>285</sup> J.B. DeLisio, X. Hu, T. Wu, G.C. Egan, G. Young, and M.R. Zachariah, J. Phys. Chem. B **120**, 5534 (2016).
- <sup>286</sup> D.F. Swinehart, J. Chem. Educ. **39**, 333 (1962).
- <sup>287</sup> H.J. Kitchen, S.R. Vallance, J.L. Kennedy, N. Tapia-Ruiz, L. Carassiti, A.

- Harrison, A.G. Whittaker, T.D. Drysdale, S.W. Kingman, and D.H. Gregory, Chem. Rev. **114**, 1170 (2014).
- <sup>288</sup> J.L. Rankinen, MICROWAVE CHARACTERIZATION OF THIN FILM TITANIA, The Pennsylvania State University, 2005.
- <sup>289</sup> C. Gong, J. Zhang, C. Yan, X. Cheng, J. Zhang, L. Yu, Z. Jin, and Z. Zhang, J. Mater. Chem. **22**, 3370 (2012).
- <sup>290</sup> F. Noor, A. Vorozhtsov, M. Lerner, and D. Wen, Powder Technol. **282**, 19 (2015).
- <sup>291</sup> R.K. Sharma, R. Kaneriya, S. Patel, A. Bindal, and K.C. Pargaien, Microelectron. Eng. **108**, 45 (2013).
- <sup>292</sup> R.A. Chandru, N. Balasubramanian, C. Oommen, and B.N. Raghunandan, J. Propuls. Power **34**, 1090 (2018).
- <sup>293</sup> F.D. Ruz-Nuglo and L.J. Groven, Adv. Eng. Mater. **20**, 1700390 (2018).
- <sup>294</sup> D.T. Zimmerman, J.D. Cardellino, K.T. Cravener, K.R. Feather, N.M. Miskovsky, and G.J. Weisel, Appl. Phys. Lett. **93**, 214103 (2008).
- <sup>295</sup> G. Young, H. Wang, and M.R. Zachariah, Propellants, Explos. Pyrotech. **40**, 413 (2015).
- <sup>296</sup> A.D. Light, (2013).

## List of Publications

Below is a list of peer-reviewed scholarly publications that I have been involved with at the time of my dissertation (including those that are under review).

\* Denotes co-first author.

1. R.J. Jacob\*, **D.J. Kline\***, M.R. Zachariah, High speed 2-dimensional temperature measurements of nanothermite composites: Probing thermal vs. Gas generation effects, *J. Appl. Phys.* 123 (2018) 115902. doi:10.1063/1.5021890.
2. H. Wang, **D.J. Kline**, M. Rehwoldt, M.R. Zachariah, Ignition and combustion characterization of Ca(IO<sub>3</sub>)<sub>2</sub>-based pyrotechnic composites with B, Al, and Ti, *Propellants, Explos. Pyrotech.* 43 (2018) 977–985. doi:10.1002/prep.201800041.
3. P.M. Guerieri, R.J. Jacob, **D.J. Kline**, A. Kerr, D. Mayo, E.E. Foos, M.R. Zachariah, Triisobutylaluminum additive for liquid hydrocarbon burn enhancement, *Combust. Flame.* 200 (2019) 53–59. doi:10.1016/j.combustflame.2018.09.023.
4. H. Wang, M. Rehwoldt, **D.J. Kline**, T. Wu, P. Wang, M.R. Zachariah, Comparison study of the ignition and combustion characteristics of directly-written Al/PVDF, Al/Viton and Al/THV composites, *Combust. Flame.* 201 (2019) 181–186. doi:10.1016/j.combustflame.2018.12.031.
5. Y. Qiao, S. Xu, Y. Liu, J. Dai, H. Xie, Y. Yao, X. Mu, C. Chen, **D.J. Kline**, E.M. Hitz, B. Liu, J. Song, P. He, M.R. Zachariah, L. Hu, Transient, in situ synthesis of ultrafine ruthenium nanoparticles for a high-rate Li–CO<sub>2</sub> battery, *Energy Environ. Sci.* 12 (2019) 1100–1107. doi:10.1039/C8EE03506G.
6. H. Wang, **D.J. Kline**, M. Rehwoldt, T. Wu, W. Zhao, X. Wang, M.R. Zachariah, Architecture can significantly alter the energy release rate from nanocomposite energetics, *ACS Appl. Polym. Mater.* 1 (2019) 982–989. doi:10.1021/acsapm.9b00016.
7. H. Wang, J. Shen, **D.J. Kline**, N. Eckman, N.R. Agrawal, T. Wu, P. Wang, M.R. Zachariah, Direct writing of a 90 wt% particle loading nanothermite, *Adv. Mater.* 31 (2019) 1–7. doi:10.1002/adma.201806575.
8. P. Ghildiyal, Y. Yang, **D.J. Kline**, S. Holdren, M.R. Zachariah, Ultrafast, scalable laser photothermal synthesis and writing of uniformly dispersed metal nanoclusters in polymer films, *Nanoscale.* 11 (2019) 13354–13365. doi:10.1039/C9NR02839K.
9. H. Wang\*, **D.J. Kline\***, M.R. Zachariah, In-operando high-speed microscopy and thermometry of reaction propagation and sintering in a nanocomposite, *Nat. Commun.* 10 (2019) 3032. doi:10.1038/s41467-019-10843-4.
10. Y. Yao, Z. Huang, P. Xie, T. Li, S.D. Lacey, M. Jiao, H. Xie, K.K. Fu, R.J. Jacob, **D.J. Kline**, Y. Yang, M.R. Zachariah, C. Wang, R. Shahbazian-Yassar, L. Hu, Ultrafast, controllable synthesis of sub-nano metallic clusters through defect engineering, *ACS Appl. Mater. Interfaces.* 11 (2019) 29773–29779. doi:10.1021/acsami.9b07198.
11. Y. Yao, Z. Huang, P. Xie, L. Wu, L. Ma, T. Li, Z. Pang, M. Jiao, Z. Liang, J. Gao, Y. He, **D.J. Kline**, M.R. Zachariah, C. Wang, J. Lu, T. Wu, T. Li, C. Wang, R. Shahbazian-Yassar, L. Hu, High temperature shockwave stabilized single atoms, *Nat. Nanotechnol.* 14 (2019) 851–857. doi:10.1038/s41565-019-0518-7.
12. S. Xu, G. Zhong, C. Chen, M. Zhou, **D.J. Kline**, R.J. Jacob, H. Xie, S. He, Z. Huang, J. Dai, A.H. Brozena, R. Shahbazian-Yassar, M.R. Zachariah, S.M. Anlage, L. Hu, Uniform, scalable, high-temperature microwave shock for nanoparticle synthesis through defect engineering, *Matter.* 1 (2019) 759–769. doi:10.1016/j.matt.2019.05.022.
13. **D.J. Kline**, M.C. Rehwoldt, H. Wang, N.E. Eckman, M.R. Zachariah, Why does adding a poor thermal conductor increase propagation rate in solid propellants?, *Appl. Phys. Lett.* 115 (2019) 114101. doi:10.1063/1.5113612.
14. G. Zhong, S. Xu, C. Chen, **D.J. Kline**, M. Giroux, Y. Pei, M. Jiao, D. Liu, R. Mi, H. Xie, B. Yang, C. Wang, M.R. Zachariah, L. Hu, Synthesis of metal oxide nanoparticles by rapid, high-temperature 3D microwave heating, *Adv. Funct. Mater.* (2019) 1904282. doi:10.1002/adfm.201904282.

15. P.M. Guerieri, R.J. Jacob, H. Wang, **D.J. Kline**, M.R. Zachariah, Droplet combustion of kerosene augmented by stabilized nanoaluminum/oxidizer composite mesoparticles, *Combust. Flame.* 211 (2020) 1–7. doi:10.1016/j.combustflame.2019.07.031.
16. M.C. Rehwoldt, H. Wang, **D.J. Kline**, T. Wu, N. Eckman, P. Wang, N.R. Agrawal, M.R. Zachariah, Ignition and combustion analysis of direct write fabricated aluminum/metal oxide/PVDF films, *Combust. Flame.* 211 (2020) 260–269. doi:10.1016/j.combustflame.2019.08.023.
17. J. Shen, Z. Liu, B. Xu, F. Chen, Y. Zhu, Y. Fu, **D.J. Kline**, X. Liao, Z. Wang, Tuning the thermal, mechanical, and combustion properties of NC-TEGDN-RDX propellants via incorporation of graphene nanoplates, *J. Energ. Mater.* (2019) 1–10. doi:10.1080/07370652.2019.1689314.
18. W. Zhao, X. Wang, H. Wang, T. Wu, **D.J. Kline**, M. Rehwoldt, H. Ren, M.R. Zachariah, S. Holdren, M. Rehwoldt, H. Ren, M.R. Zachariah, Titanium enhanced ignition and combustion of Al/I<sub>2</sub>O<sub>5</sub> mesoparticle composites, *Combust. Flame.* 212 (2020) 245–251. doi:10.1016/j.combustflame.2019.04.049.
19. X. Wang, Z. Huang, Y. Yao, H. Qiao, G. Zhong, Y. Pei, C. Zheng, **D.J. Kline**, Q. Xia, Z. Lin, J. Dai, M.R. Zachariah, B. Yang, R. Shahbazian-Yassar, L. Hu, Continuous 2000 K droplet-to-particle synthesis, *Mater. Today.* (2020). doi:10.1016/j.mattod.2019.11.004.
20. **D.J. Kline\***, M.C. Rehwoldt\*, C.J. Turner, P. Biswas, G.W. Mulholland, S.M. McDonnell, M.R. Zachariah, Spatially focused microwave ignition of metallized energetic materials, *J. Appl. Phys.* 127 (2020) 55901. doi:10.1063/1.5134089.
21. J. Shen, H. Wang, **D.J. Kline**, Y. Yang, X. Wang, M. Rehwoldt, T. Wu, S. Holdren, M.R. Zachariah, Combustion of 3D printed 90 wt% loading reinforced nanothermite, *Combust. Flame.* 215 (2020) 86–92. doi:10.1016/j.combustflame.2020.01.021.
22. Y. Yang, Y. Yao, **D.J. Kline**, T. Li, P. Ghildiyal, H. Wang, L. Hu, M.R. Zachariah, Rapid laser pulse synthesis of supported metal nanoclusters with kinetically tunable size and surface density for electrocatalytic hydrogen evolution, *ACS Appl. Nano Mater.* (2020). doi:10.1021/acsanm.0c00238.
23. **D.J. Kline\***, Z. Alibay\*, M.C. Rehwoldt, A. Idrogo-Lam, S.G. Hamilton, P. Biswas, F. Xu, M.R. Zachariah, Experimental observation of the heat transfer mechanisms that drive propagation in additively manufactured energetic materials, *Combust. Flame.* 215 (2020) 417–424. doi:10.1016/j.combustflame.2020.01.020.
24. P. Biswas, G. Mulholland, M. Rehwoldt, **D.J. Kline**, M.R. Zachariah, Microwave absorption by small dielectric and semi-conductor coated metal particles, *J. Quant. Spectrosc. Radiat. Transf.* 247 (2020) 106938. doi:10.1016/j.jqsrt.2020.106938.
25. M. Wu, M. Cui, L. Wu, S. Hwang, C. Yang, Q. Xia, G. Zhong, H. Qiao, W. Gan, X. Wang, **D.J. Kline**, M.R. Zachariah, D. Su, T. Li, L. Hu, Hierarchical polyelemental nanoparticles as bifunctional catalysts for oxygen evolution and reduction reactions, *Adv. Energy Mater.* (2020) 2001119. doi:10.1002/aenm.202001119.
26. G. Zhong, C. Wang, R. Wang, W. Ping, S. Xu, H. Qiao, M. Cui, X. Wang, Y. Zhou, **D.J. Kline**, M.R. Zachariah, L. Hu, Rapid, high-temperature microwave soldering toward a high-performance cathode/electrolyte interface, *Energy Storage Mater.* (2020). doi:10.1016/j.ensm.2020.05.015.
27. H. Wang, B. Julien, **D.J. Kline**, Z. Alibay, M.C. Rehwoldt, C. Rossi, M.R. Zachariah, Probing the reaction zone of nanolaminates at ~μs time and ~μm spatial resolution, *J. Phys. Chem. C.* 124 (2020) 13679–13687. doi:10.1021/acs.jpcc.0c01647.
28. W. Zhao, H. Ren, T. Yan, Y. Ou, Q. Jiao, H. Wang, **D.J. Kline**, M.R. Zachariah, Tailoring energy release of nano-Si based thermites via incorporation of Ti nanoparticles, *Chem. Eng. J.* 396 (2020) 124559. doi:10.1016/j.cej.2020.124559.
29. M.C. Rehwoldt\*, **D.J. Kline\***, M.R. Zachariah, Numerically evaluating energetic composite flame propagation with thermally conductive, high aspect ratio fillers, *Chem. Eng. Sci.* 229 (2021) 116087. doi:10.1016/j.ces.2020.116087. *Featured cover article.*
30. H. Xie, M. Hong, E.M. Hitz, X. Wang, M. Cui, **D.J. Kline**, M.R. Zachariah, L. Hu, High-Temperature Pulse Method for Nanoparticle Redispersal, *J. Am. Chem. Soc.* (2020). doi:10.1021/jacs.0c04887.
31. F. Xu, P. Biswas, G. Nava, J. Schwan, **D.J. Kline**, M.C. Rehwoldt, L. Mangolini, M.R. Zachariah, Tuning the reactivity and energy release rate of I<sub>2</sub>O<sub>5</sub> based ternary thermite systems, *Under*

- review.*
32. P. Ghildiyal, X. Ke, G. Nava, **D.J. Kline**, H. Wang, L. Mangolini, M.R. Zachariah, Silicon nanoclusters for reactivity and energy density enhancement of biocidal energetic mesoparticle composites, *Under review*.
  33. Z. Alibay\*, **D.J. Kline\***, M.C. Rehwoldt, P. Biswas, S. Herrera, M.R. Zachariah, Mechanism of microwave-initiated ignition of sensitized energetic nanocomposites, *Under review*.
  34. Z. Liang, Y. Yao, B. Jiang, X. Wang, H. Xie, M. Jiao, C. Liang, H. Qiao, **D.J. Kline**, M.R. Zachariah, L. Hu, 3D printed 3000 K probe, *Under review*.
  35. **D.J. Kline**, M.C. Rehwoldt, J.B. DeLisio, S.C. Barron, H. Wang, Z. Alibay, J.C. Rodriguez, G.M. Fritz, M.R. Zachariah, In-operando thermophysical property measurements and their relationship to propagation in additively manufactured Al-Zr-C composites, *Under review*.
  36. H. Xie, Y. Liu, N. Li, W. Ping, S.D. Lacey, C. Yang, Y. Zhou, **D.J. Kline**, Y. Yao, M. Jiaou, B. Liu, S. He, M.R. Zachariah, D. Su, C. Wang, L. Hu, High temperature pulse synthesis of ultrathin graphene coated metal nanoparticles for electrocatalytic applications, *Under review*.
  37. H. Wang, **D.J. Kline**, P. Biswas, M.R. Zachariah, Imaging agglomeration in a thermite reaction: Role of oxidizer size and morphology on propagation, *Under review*.